



# ADMINISTRAÇÃO DE DADOS - PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS

N/AD/0001/05

Data: 25/02/2019



ADMINISTRAÇÃO DE DADOS

NORMA

PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS

**Nota da versão:**

Versão 05 – Alteração do nome da Norma de Administração de Dados – Padrões e Nomenclaturas” para “Padrões para construção de Modelos de Dados”; Alteração para consideração de nova tecnologia *Hadoop* e processos de Governança, envolvendo o conceito de *Data Lake*.



**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

**SUMÁRIO**

- 1 OBJETIVO**
- 2 APLICAÇÃO**
- 3 ÓRGÃOS CITADOS NA NORMA**
- 4 CONCEITUAÇÃO**
  - 4.1 Gestão de Dados**
  - 4.2 Administração de Dados**
  - 4.3 Administrador de Dados – AD**
  - 4.4 Administrador de Banco de D – DBA**
  - 4.5 Analista Especificador**
  - 4.6 Modelagem de Dados**
    - 4.6.1 Modelo Lógico de Dados**
      - 4.6.1.1 Entidade**
      - 4.6.1.2 Atributo**
      - 4.6.1.3 Relacionamento**
        - 4.6.1.3.1 Cardinalidade**
      - 4.6.1.4 Restrição de Integridade**
      - 4.6.1.5 Relação Normalizada**
    - 4.6.2 Modelo Físico de Dados**
  - 4.7 Conceitos relativos a Tecnologias de Armazenamento para Aplicações Analíticas**
    - 4.7.1 Modelagem de Dados Multidimensional**
    - 4.7.2 Data Warehouse (DW)**
    - 4.7.3 Data Mart (DM)**
    - 4.7.4 Operational Data Store (ODS)**



**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

- 4.7.5 Staging Area (SA)
- 4.7.6 Área de Trabalho (AT)
- 4.7.7 Data Lake
- 4.7.8 Tabela Fato
- 4.7.9 Tabela Dimensão
- 4.7.10 Tabela Ponte
- 4.7.11 Modelagem de Dados - Esquema Estrela (Star schema)
- 4.7.12 Modelagem de Dados - Esquema Floco de Neve (Snowflake schema)
- 4.8 Gerenciamento de Versão
  - 4.8.1 Projeto de Dados
  - 4.8.2 Baseline
  - 4.8.3 Stream
  - 4.8.4 Snapshot
- 4.9 Modelo de Dados Corporativos e Compartilhados
- 5 DIRETRIZES BÁSICAS
  - 5.1 Nomenclatura para Projetos de Dados
  - 5.2 Nomenclatura dos Modelos de Dados
    - 5.2.1 Nomenclatura dos Modelos de Dados Lógicos
    - 5.2.2 Nomenclatura dos Modelos de Dados Físicos
  - 5.3 Nomenclatura das Streams
  - 5.4 Nomenclatura para Baselines e Snapshots
  - 5.5 Especificação de Diagramas
  - 5.6 Especificação de entidades e tabelas
    - 5.6.1 Prefixo de identificação de tabelas com tipo específico
  - 5.7 Especificação dos atributos e colunas
  - 5.8 Tipos de dados para SGBD Oracle



**ADMINISTRAÇÃO DE DADOS**

**NORMA**

**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

- 5.9** Tipos de dados para SGBD PostgreSQL
- 5.10** Nomenclatura para demais elementos nos modelos de dados
  
- 6** **PROCESSO DE INTEGRAÇÃO DE MODELOS DE DADOS**
- 6.1** Processo de Integração nas Aplicações Analíticas
  
- 7** **ARQUITETURA DE DADOS PARA APLICAÇÕES ANALÍTICAS**
  
- 8** **VIGÊNCIA**

**Elementos Complementares:**

**Anexo I**                    **Referências**



**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

**1 OBJETIVO**

Estabelecer diretrizes, padrões e nomenclaturas para a construção de modelos de dados.

**2 APLICAÇÃO**

Aplica-se a todos os órgãos da Empresa.

**3 ÓRGÃOS CITADOS NA NORMA**

<b>Sigla</b>	<b>Função principal</b>
COGD	Órgão responsável pela Gestão de Dados

**4 CONCEITUAÇÃO**

**4.1 Gestão de Dados**

É a função na organização que cuida do planejamento, controle e entrega de ativos de dados e de informação. Esta função inclui as disciplinas do desenvolvimento, execução e supervisão de planos, políticas, programas, projetos, processos, práticas e procedimentos que controlam, protegem, distribuem e aperfeiçoam o valor dos ativos de dados e informações. (DAMA-DMBOK®)

**4.2 Administração de Dados**

É a função responsável por desenvolver e administrar de modo centralizado as estratégias, procedimentos e práticas para o processo de gerência dos recursos de dados e aplicativos, incluindo planos para a sua definição, padronização, organização, proteção e utilização. (BARBIERI, Carlos)

**4.3 Administrador de Dados – AD**

Responsável por estabelecer e disseminar padrões de administração de dados, políticas, boas práticas e procedimentos para a construção e validação dos modelos de dados da empresa. Também é responsável por modelar ou apoiar a construção e/ou alteração dos modelos de dados conceituais, lógicos e físicos, assim como pela implementação e atualização das estruturas de dados nos bancos de dados em ambiente de desenvolvimento.



**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

**4.4 Administrador de Banco de Dados– DBA**

É o responsável pela criação e manutenção das estruturas de dados nos bancos de dados de missão crítica como homologação e produção. Também é responsável pela criação, manutenção e monitoração dos bancos de dados, suporte aos seus usuários, bem como tomar as ações necessárias para garantir a integridade e disponibilidade destes ambientes.

**4.5 Analista Especificador**

Analista que especifica os casos de uso em contato com o cliente e participa da construção do modelo de dados junto ao Administrador de Dados.

**4.6 Modelagem de Dados**

A modelagem de dados pode ser definida como um processo onde, através do levantamento dos requisitos de informação e regras de negócio, aplicando técnicas e mecanismos de abstração, construímos artefatos (modelos de dados) que tem por objetivo representar um conjunto de informações. Estes normalmente são geridos por uma aplicação.

**4.6.1 Modelo Lógico de Dados**

É aquele em que os objetos, suas características e relacionamentos não seguem, necessariamente, as regras de implementação limitantes impostas por algum tipo de tecnologia de armazenamento, embora já seja possível definir algumas de suas propriedades físicas, levando-se em consideração, características específicas de um Sistema Gerenciador de Banco de Dados (SGBD) por exemplo. Neste nível, respeita-se conceitos de chaves de acesso, normalização, integridade referencial, formatos de campos, etc.

O modelo lógico de dados deve ter uma disposição correta e bem distribuída, tanto das entidades quanto dos relacionamentos, sendo este um critério a ser observado na avaliação do modelo

**4.6.1.1 Entidade**

É a representação de “algo” que existe no mundo real. Uma entidade pode representar um elemento concreto, como uma pessoa ou um livro, ou um elemento abstrato, como um empréstimo, uma viagem ou um conceito. Um conjunto de entidades contém elementos que são similares e compartilham as mesmas características. Cada elemento deste conjunto é dito uma ocorrência da entidade e será identificado de forma unívoca em relação aos outros elementos. As entidades são classificadas em:

- **Entidade Associativa:** A entidade associativa surge de um relacionamento NxM, em que existe uma associação dos atributos identificadores das duas entidades relacionadas, caracterizando uma nova entidade. A nova entidade gerada possui, normalmente, atributos próprios do relacionamento. Também se utiliza esse tipo de entidade para representar relacionamentos múltiplos (envolvendo mais de 2 entidades) ou agregações (relacionamentos que se relacionam com outras entidades).

### PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS

- **Entidade Forte (Independente ou Dominante):** uma entidade x é dita forte quando sua existência não depende da existência de outra entidade.
- **Entidade Fraca (Dependente ou Subordinada):** uma entidade x é chamada fraca quando sua existência depende hierarquicamente da existência de outra entidade y. A entidade y é chamada entidade dominante (forte) e a x é chamada entidade subordinada (fraca).
- **Superclasse (Supertipo):** quando diferentes entidades possuem características comuns, essas características podem ser agrupadas em uma entidade genérica de nível superior chamada superclasse. O processo de agrupamento é chamado generalização.
- **Subentidade (Subtipo):** quando a partir da subdivisão de uma entidade, segundo um critério de classificação, são obtidas outras entidades com características próprias. Cada entidade obtida com a subdivisão é chamada de subclasse e representa um subconjunto da entidade superior, a superclasse. O processo de subdivisão é dito Especialização.

#### 4.6.1.2 Atributo

Uma entidade é representada por um conjunto de atributos. Um atributo é uma propriedade que qualifica, descreve ou identifica um dado dentro da entidade. Os atributos descrevem cada membro de um conjunto de entidades. Um atributo pode ser:

- **Simple (Atômico):** quando o atributo possui um valor que não é divisível (ex.: sexo);
- **Composto:** quando o atributo possui um valor que pode ser dividido em partes menores (outros atributos) (ex.: endereço);
- **Monovalorado:** quando o atributo assume um único valor para cada elemento do conjunto de entidades (ex.: nome);
- **Multivalorado:** quando o atributo pode ter mais de um valor para uma única entidade (ex.: dependentes, telefones);
- **Derivado:** quando o valor do atributo é derivado de outros atributos da própria entidade ou a partir de atributos de entidades relacionadas (ex.: idade);
- **Nulo:** em alguns casos, uma determinada entidade pode não ter um valor aplicável para um atributo, ou este valor é desconhecido (ex.: número do apartamento).

#### 4.6.1.3 Relacionamento

Trata-se da associação existente entre uma ou mais entidades. As modalidades de relacionamentos são as seguintes:

- **Relacionamento Identificador:** é aquele que faz parte da composição do identificador único de uma entidade.



**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

- **Relacionamento Não Identificador:** é aquele que não faz parte da composição do identificador único de uma entidade.
- **Auto-relacionamento:** é aquele em que uma entidade se relaciona com ela mesma (relacionamento recursivo).

**4.6.1.3.1 Cardinalidade**

É a quantidade de vezes em que um elemento de um conjunto de entidades pode, em um determinado instante, estar associado, em um dado relacionamento, a outros elementos de outras entidades. Quanto à cardinalidade, os relacionamentos podem ser classificados como:

- **Um para um (1:1):** Quando um elemento da entidade A pode estar associado no máximo a um elemento da entidade B, e um elemento da entidade B pode estar associado no máximo a um elemento da entidade A.
- **Um para muitos (1:N):** Quando um elemento da entidade A pode estar associado a vários elementos da entidade B. Entretanto, um elemento da entidade B pode estar associado no máximo a um elemento da entidade A.
- **Muitos para muitos (N:M):** Quando um elemento da entidade A pode estar associado a vários elementos da entidade B, e um elemento da entidade B pode estar associado a vários elementos da entidade A.

**4.6.1.4 Restrição de Integridade**

É uma regra que deve ser obedecida para que se mantenha a consistência dos dados de um banco de dados. As restrições de integridade podem ser dos seguintes tipos:

- **Conteúdo:** quando um determinado atributo deve obrigatoriamente possuir um valor ou não. É o que definimos de atributo obrigatório ou opcional.
- **Domínio:** quando limita os valores que um atributo pode conter. Fica registrada quando se define o formato, tamanho, e valores possíveis de um atributo.
- **Referencial:** representada por um relacionamento entre duas entidades do modelo lógico, é usada para manter a consistência entre os dados relacionados nas duas entidades.
- **Negócio (Semântica):** quando representa uma regra característica do negócio em estudo, que não se enquadra nos demais tipos.

**4.6.1.5 Relação Normalizada**

- A normalização de dados consiste na realização de vários passos seguidos no projeto de um banco de dados, que permitem um armazenamento consistente e eficiente acesso aos dados em bancos de dados relacionais. Esses passos reduzem a redundância de dados e as chances dos dados se tornarem inconsistentes.

**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

- A técnica de normalização prevê a existência de regras, chamadas de formas normais, que devem ser aplicadas com o objetivo de alcançar certas condições desejáveis que uma relação deve satisfazer.
- Uma relação normalizada é a que esteja pelo menos na terceira forma normal, ou seja:
- Não contenha colunas com elementos multivalorados ou compostos (1ª forma normal);
- Cada coluna que não faça parte da chave primária depende, funcionalmente, de todas as partes da chave primária (2ª forma normal);
- Cada coluna que não faça parte da chave primária depende, funcionalmente, apenas da chave primária, e de nenhuma outra coluna fora da chave primária (3ª forma normal).

**4.6.2 Modelo Físico de Dados**

O modelo de dados físico deve ser criado a partir da transformação de um modelo de dados lógico. Nele a representação dos objetos, suas características e relacionamentos seguem as regras de implementação impostas pelo tipo de tecnologia de armazenamento utilizada para implementar as estruturas de dados deste. Neste nível, são introduzidos conceitos clássicos relacionados a modelos de dados físicos estruturados.

- **Tabela:** estrutura de armazenamento de dados, formada por um conjunto finito de colunas e ilimitado de linhas (tuplas). Uma tabela é equivalente a uma entidade no modelo lógico.
- **Linha (ou tupla):** representação equivalente à ocorrência de uma entidade. Cada linha representa um registro dentro da tabela.
- **Coluna:** são unidades básicas de armazenamento de dados em uma linha dentro de uma tabela. Uma coluna é similar a um atributo no modelo lógico.
- **Chave Primária:** é um atributo ou um conjunto de atributos cujos valores distinguem ou identificam de forma unívoca cada linha dentro de uma tabela.
- **Chave Única (Unique Key):** elemento que especifica que outros valores de coluna(s), além da chave primária, devem ser únicos.
- **Chave Estrangeira (Foreign Key):** representa um atributo, ou combinações de atributos, cujos valores aparecem na chave primária ou em uma chave única de outra entidade que se relaciona com esta, onde encontramos a chave estrangeira. A chave estrangeira é oriunda do relacionamento entre as entidades.
- **Índice:** recurso físico que visa otimizar a recuperação de uma informação, via um método de acesso, fornecendo acesso rápido a linhas específicas no banco de dados, eliminando a necessidade de varreduras completas de tabelas. Um índice é a seleção de uma ou mais colunas de uma tabela para já armazenar os seus dados ordenados,

**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

acompanhado de ponteiros que servirão como atalhos para a localização física dos registros.

- **Regra de Verificação (Check Constraint):** elemento que especifica regras para os valores que uma coluna poderá receber.
- **Sequência (Sequence):** elemento que gera uma numeração serial e única para colunas numéricas de uma tabela.
- **Visão (View):** Representa uma visão parcial ou completa de uma ou mais tabelas.
- **Visão Materializada (Materialized View):** réplica de uma tabela ou visão a partir de um único ponto no tempo. Muito utilizada em modelos multidimensionais para compartilhar estruturas de dados.

#### 4.7 Conceitos relativos a Tecnologias de Armazenamento para Aplicações Analíticas

##### 4.7.1 Modelagem de Dados Multidimensional

É uma técnica de construção de modelos de dados na qual a informação reside na interseção de várias dimensões. A modelagem multidimensional permite que o usuário perceba os dados em uma forma próxima de seu entendimento, com várias perspectivas possíveis. É aplicada na elaboração de modelos de dados que visam atender aos objetivos principais de projetos de *Data Warehouse* (DW) e *Data Mart* (DM). Estes devem proporcionar meios que facilitem a investigação, resumo e organização de dados para a análise, resultando em relatórios de apoio à tomada de decisões gerenciais e estratégicas.

##### 4.7.2 Data Warehouse (DW)

É um tipo de banco de dados voltado ao apoio à tomada de decisões gerenciais e estratégicas, cuja tecnologia visa promover melhores negócios à empresa a partir da análise de grande quantidade de informação que se encontra distribuída por diversos sistemas de produção e sistemas externos. O projeto e a implementação de um DW requerem a utilização de conceitos, técnicas, metodologias e ferramentas adicionais das utilizadas nos bancos de dados transacionais.

##### 4.7.3 Data Mart (DM)

É um subconjunto departamental de um DW, concentrado em assuntos de negócio (Pedidos, Vendas, Finanças, Concessão de Benefícios aos filiados da Previdência Social, entre outros), ou seja, o DM é um DW com escopo de projeto delimitado a uma área ou assunto específicos de uma empresa.

**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS****4.7.4 Operational Data Store (ODS)**

Um *Operational Data Store (ODS)* é um conjunto de dados baseado em assuntos, integrado, volátil, atual ou recente, distinto do *Data Warehouse*, de apoio às decisões operacionais do dia a dia.

**4.7.5 Staging Area (SA)**

Área ou local destinado a representar as entidades clonadas a partir das fontes de dados de origem e/ou entidades utilizadas para tratamento das informações necessárias para as operações de ETL.

**4.7.6 Área de Trabalho (AT)**

Conceito similar à SA adotado na Dataprev.

**4.7.7 Data Lake**

*Data Lake*, em essência, é uma estratégia de armazenamento de dados. O *Data Lake* pode armazenar dados estruturados e não estruturados em seu formato bruto e são projetados para o consumo de dados, apoiando a descoberta de novas perguntas sobre os dados. Como o *Data Lake* em sua grande maioria armazena os dados em seu formato bruto, cabe àqueles que vão analisar os dados, dar sentido a esses dados para o propósito ao qual a análise se destina. Na Dataprev, este também armazena dados trabalhados/tranformados a partir dos dados brutos previamente carregados neste, com o objetivo de aproximar o dado do resultado esperado a ser analisado pelo cliente.

**4.7.8 Tabela Fato**

É uma estrutura que armazena medidas associadas aos eventos de negócio. Cada uma das métricas é obtida pela interseção de todas as dimensões que compõem o esquema estrela. Ela possui uma considerável quantidade de linhas (maior volume de dados) e sua chave primária, geralmente, é constituída das chaves estrangeiras de todas as dimensões, considerando o nível de granularidade mais abrangente. A tabela fato é normalizada e esparsa, dado a grande quantidade de combinações possíveis.

**4.7.9 Tabela Dimensão**

Estrutura que armazena as descrições textuais das entidades representativas do negócio, tais como tempo, localização geográfica, produto, cliente etc. Possuem uma relação de 1:N com a tabela fato. Ela dispõe de um número significativamente menor de linhas cujas colunas (atributos) são textuais e algumas delas estão dispostas hierarquicamente.

**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS****4.7.10 Tabela Ponte**

Tabela com uma chave composta que captura um relacionamento de muitos para muitos (n para m) que não pode ser acomodado pela granularidade peculiar de uma tabela de fatos simples ou de uma tabela de uma única dimensão. Funciona como ponte entre a tabela de fatos e a tabela de dimensões para aceitar dimensões com diversos valores ou hierarquias desiguais. Também conhecida como tabela associativa ou auxiliar.

**4.7.11 Modelagem de Dados - Esquema Estrela (*Star schema*)**

Método de modelagem de dados, no qual os dados são modelados em tabelas dimensionais ligadas a uma tabela de fatos. As tabelas dimensionais contêm as características de um evento. A tabela de fatos armazena os fatos ocorridos e as chaves para as características correspondentes, nas tabelas dimensionais.

O nome foi adotado devido à semelhança do modelo com uma estrela. No “centro” da estrela, existem as tabelas de fatos, rodeadas por tabelas auxiliares, chamadas de dimensões.

**4.7.12 Modelagem de Dados - Esquema Floco de Neve (*Snowflake schema*)**

Método de modelagem de dados que representa uma variação do esquema estrela, porém possui ao menos uma de suas tabelas de dimensão normalizadas.

**4.8 Gerenciamento de Versão****4.8.1 Projeto de Dados**

Conjunto de modelos de dados lógico com alto grau de dependência entre si e seus modelos de dados físicos derivados. Usualmente este alto grau de dependência indica que suas alterações precisam estar alinhadas, indicando que estas em conjunto devem conceitualmente compor uma versão. Representa um conceito mais amplo do que um Projeto de Banco de Dados, uma vez que um Projeto de Dados pode representar estruturas de dados que não necessariamente são persistidas em Bancos de Dados Objeto Relacionais, podendo ser persistidas em estruturas de dados não convencionais.

**4.8.2 *Baseline***

Representa uma versão de um Projeto de Dados. Complementarmente, segundo a IEEE (IEEE Std. No. 610.12-12-1990), é uma especificação ou um produto que foi formalmente revisado e acordado, e que depois disso serve como base para futuros desenvolvimentos, e que podem ser alterados somente através de um processo formal de controle de alterações.

**4.8.3 *Stream***

**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

É um objeto do repositório da ferramenta IBM Rational Team Concert utilizado para integrar os trabalhos realizado pelas equipes que participam no desenvolvimento do projeto. Representa um fluxo ou linha de trabalho.

**4.8.4 Snapshot**

Representa uma versão de uma stream.

**4.9 Modelo de Dados Corporativos e Compartilhados**

Modelos de Dados Corporativos e Modelos de Dados Compartilhados são ferramentas de integração que auxiliam na conexão entre modelos de dados relacionados, servindo de ponto de integração e reuso. Os Modelos de Dados Corporativos são modelos de dados onde se agrupam entidades reconhecidas por diversas linhas de negócio da Dataprev. Por sua vez os Modelos de Dados Compartilhados se diferenciam dos Corporativos apenas por possuírem um escopo mais específico, usualmente sendo restrito a assuntos de um cliente ou linha de negócio. Desta forma, o principal objetivo deste tipo de ferramenta de integração é permitir a entrada de dados por um único modelo de dados e o compartilhamento desta informação com todos os sistemas que necessitem dela, sem a necessidade de manter sincronizações entre dados em diferentes sistemas.

**5 DIRETRIZES BÁSICAS**

Os elementos do modelo de dados devem estar padronizados de acordo com esta norma.

O modelo de dados pode ser construído em conjunto por analistas de negócios, analistas especificadores e administradores de dados.

O modelo de dados deve ser construído buscando a integração com os demais modelos de dados do cliente, evitando-se redundâncias.

O modelo de dados a ser construído deve se utilizar de elementos de dados do Modelo de Dados Corporativos e Compartilhados, quando couber.

O modelo de dados deve ser construído prezando pelas boas práticas da tecnologia na qual será implementado

**5.1 Nomenclatura para Projetos de Dados**

Deve acompanhar a sigla do nome do produto que está sendo desenvolvido. Este pode ser subdividido em mais de um modelo lógico se possuir pacotes ou funcionalidades distintas dentro do mesmo produto.

**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

Os Projetos de Dados feitos para aplicações analíticas receberão o prefixo “BG\_” para identificá-lo como uma “Base de Gestão” e um sufixo para classificá-la. A saber:

- BG\_...\_DW: Para projeto de Data Warehouse.
- BG\_...\_DM: Para projeto de Data Mart, dependente ou não do Data Warehouse.
- BG\_...\_ODS: Para projeto de Operational Data Store(ODS).
- BG\_...\_SA: Para projeto contendo apenas tabelas de Staging Area e Área de Trabalho.
- BG\_...\_QVD : para projeto em Qlik View cujos repositórios sejam arquivos QVD.

Por sua vez, Projetos de Dados para Data Lake receberão o prefixo conforme abaixo:

- DL\_...: Para projeto do Data Lake

**5.2 Nomenclatura dos Modelos de Dados****5.2.1 Nomenclatura dos Modelos de Dados Lógicos**

Para criação de modelos de dados lógicos na ferramenta CASE, devem ser considerados os seguintes padrões de denominação:

**<NomeDaAplicação>**, onde:

- **<NomeDaAplicação>**: Nome da Aplicação;

**5.2.2 Nomenclatura dos Modelos de Dados Físicos**

Para criação de modelos de dados físicos na ferramenta CASE, devem ser considerados os seguintes padrões de denominação:

**<NomeDaAplicação>\_<CategoriaDaBaseDeDados>** onde:

- **<NomeDaAplicação>**: Nome da Aplicação;
- **<CategoriaDaBaseDeDados>**: Categoria da base de dados, que podem ser:
  - > **DEV**: Base de dados de Desenvolvimento;
  - > **HML**: Base de dados de homologação;
  - > **PRD**: Base de dados de Produção;
  - > **TRN**: Base de dados de Treinamento;
  - > **TIN**: Base de dados de Teste Integrado;
  - > **TSA**: Base de dados de Teste Automatizado;

**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

- > **TDE**: Base de dados de Teste de Desempenho;
- > **MND**: Base de dados de Manutenção em ambiente de Desenvolvimento;
- > **MNH**: Base de dados de Manutenção em ambiente de Homologação;
- > **SEG**: Base de dados de desenvolvimento seguro.

Exemplo: SIRC\_DEV

### 5.3 Nomenclatura das *Streams*

Para criação de *streams*, devem ser considerados os seguintes padrões de denominação:

**str\_<NomeDaAplicação>\_<TipoDaStream>** onde:

- **str**: Abreviatura de stream;
- **<NomeDaAplicação>**: Nome da aplicação;
- **<TipoDaStream>**: tipo da stream, que pode ser:
  - > **HEAD**: Stream principal, onde são geradas todas as Baselines e Snapshots;
  - > **DEV**: Stream de atuação dos ADs e analistas dos projetos/produtos.

### 5.4 Nomenclatura para *Baselines e Snapshots*

Para criação das baselines e snapshots, devem ser considerados os seguintes padrões de denominação:

**BSL\_<NomeDaAplicação>\_<CategoriaDaBaseDeDados>**, corresponde à baseline ou snapshot da stream head, tendo como base o modelo de dados lógico e físico da stream dev.

Onde:

- **BSL**: Abreviação de Baseline ou snapshot;
- **<NomeDaAplicação>**: Nome da Aplicação;
- **<CategoriaDaBaseDeDados>**: Categoria da base de dados onde esta baseline pode ser implantada, que podem ser:
  - > **DEV**: Base de dados de Desenvolvimento;
  - > **HML**: Base de dados de homologação;
  - > **PRD**: Base de dados de Produção;
  - > **TRN**: Base de dados de Treinamento;
  - > **TIN**: Base de dados de Teste Integrado;
  - > **TSA**: Base de dados de Teste Automatizado;
  - > **TDE**: Base de dados de Teste de Desempenho;
  - > **MND**: Base de dados de Manutenção em ambiente de Desenvolvimento;
  - > **MNH**: Base de dados de Manutenção em ambiente de Homologação;





**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

- **SEG:** Base de dados de desenvolvimento seguro.

*Observação: Quando a <CategoriaDaBaseDeDados> não é informada no nome da baseline, significa que esta baseline pode ser implantada em todas as bases de dados do produto.*

Exemplo:

BSL\_SIRC\_HML – Pode ser implantada apenas na base de dados de Homologação.

BSL\_SIRC\_HML\_PRD – Pode ser implantada apenas nas bases de dados de Homologação e Produção.

BSL\_SIRC – Pode ser implantada em todas as bases de dados do produto.

### 5.5 Especificação de diagramas

- Para manter legível o diagrama em modelos grandes, é preferencial separar em diagramas menores, contendo as tabelas por grupo de funcionalidade ou negócio;
- As tabelas do próprio modelo não serão coloridas;
- As tabelas compartilhadas do modelo de dados corporativo (MDC) devem estar em cinza;
- As demais tabelas compartilhadas de outros sistemas possuirão cores atribuídas livremente, desde que constem em legenda no diagrama.
- Todo projeto deve ter um diagrama que represente as dependências deste projeto com os demais projetos.

### 5.6 Especificação de entidades e tabelas

Na especificação de uma entidade ou tabela, devem ser observados alguns aspectos:

- **Nome:** deve ser informado um substantivo, função ou evento, no plural, em maiúsculas, sem acentuação, podendo ser simples ou composto separado pelo sublinhado (“\_”), com no máximo 30 caracteres.
  - Admite-se nome de tabelas no singular quando não existir plural do nome informado;
  - Toda entidade ou tabela deve possuir uma abreviação do nome normal, com até 5 caracteres. Esta abreviação será usada na composição dos nomes de outros elementos associados à entidade (Chaves primárias, estrangeiras, índices, etc).

**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

- **Volume inicial:** quantidade de registros que esta tabela provavelmente armazenará no primeiro ano de uso do sistema. Esta informação será usada no cálculo para dimensionar o espaço físico no banco de dados;
- **Volume final:** quantidade de registros resultante de uma estimativa de crescimento nos próximos 5 anos;
- Tabela e entidade devem possuir uma boa documentação sobre sua descrição e função dentro do modelo de dados.

**5.6.1 Prefixo de identificação de tabelas com tipo específico**

Prefixo	Descrição	Uso
LG	Tabelas de Log	OLTP
HT	Tabelas de histórico	OLTP
TP	Tabelas temporárias	OLTP
FT	Tabelas Fato	OLAP
DI	Tabelas de Dimensão	OLAP
SA	Tabelas de Staging Area	OLAP
AT	Tabelas de área de trabalho	OLAP
OD	Tabelas de ODS (Operational Data Store)	OLAP

**5.7 Especificação dos atributos e colunas**

- O nome deve expressar o que o atributo representa;
- Devem ser iniciados com duas letras correspondentes à sua classe, seguido de sublinhado (“\_”) e o nome propriamente dito, com no máximo 30 caracteres;
- Caso haja necessidade de abreviação dos termos que compõem um nome do atributo, deve-se optar por abreviar os termos menos importantes para compreensão do atributo, ou por abreviaturas que fiquem mais inteligíveis, procurando deixar os termos mais importantes na forma extensa;
- Siglas de amplo uso no ambiente dos clientes, podem ser usadas em sua forma original. Contudo deve existir a devida descrição na propriedade de descrição do atributo;
- Na nomeação de atributos e colunas, preocupar-se sempre com a coerência do

**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

significado e para que não haja redundância, como por exemplo, evitar o uso dos nomes `cs_status`, `cs_tipo`, `te_descricao`, entre outros;

- Atributos e colunas da classe IN devem estar obrigatoriamente associadas a um domínio DM\_SIM\_NAO, com valores 0 e 1, significando respectivamente, NÃO e SIM, sempre do tipo NUMBER(1).
- As classes de atributos devem ser abreviadas, utilizando-se as seguintes definições abaixo:

Classe		Descrição
<b>ID</b>	Identificador	Prefixo utilizado em atributo que servirá de identificador da tabela.
<b>CS</b>	Classificador	Representa um número finito de valores que serão incluídos no atributo. Deve sempre estar associado a um domínio obrigatoriamente.
<b>CE</b>	Classificador Externo	Representa um número finito de valores que são desconhecidos do modelo de dados, por pertencerem a fontes de dados externas à Dataprev, sob as quais não se possui gestão. Não terá domínio associado.
<b>NU</b>	Número	Expressão de quantidade, categoria ou identificação.
<b>IN</b>	Indicador	Indica um valor lógico, do tipo sim/não.
<b>DT</b>	Data	Campo que receberá valor de uma data e, opcionalmente, também uma hora.
<b>HR</b>	Hora	Campo que receberá valor de uma hora.
<b>NM</b>	Nome	Substantivo para denominar, objetos, pessoas, eventos, etc...

**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

Classe		Descrição
<b>TE</b>	Texto	Conjunto de palavras que versa sobre algo.
<b>VL</b>	Valor	O preço monetário atribuído a uma coisa; estimação, valoração.
<b>QT</b>	Quantidade	Indica o número de unidades em determinada grandeza.
<b>TX</b>	Taxa	Índice, imposto e taxa.
<b>SG</b>	Sigla	Abreviação de uma palavra ou expressão
<b>IM</b>	Imagem	figura, fotografia, etc.
<b>VD</b>	Vídeo	Vídeo
<b>SO</b>	Som	Som
<b>DO</b>	Documento	Documento em geral, como planilha, documento, PDF, etc.

**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS****5.8 Tipos de dados para SGBD Oracle**

<b>Classe</b>	<b>Tipo de Dado</b>	<b>Observações</b>
<b>ID</b>	NUMBER(X)	Especificar o tamanho do atributo de acordo com o volume máximo da tabela, acrescentando-se em média dois dígitos conforme a expectativa de crescimento
<b>CS</b>	NUMBER(X) VARCHAR2(X)	Especificar o valor máximo do atributo
<b>CE</b>	NUMBER(X) VARCHAR2(X)	Especificar o valor máximo do atributo
<b>NU</b>	NUMBER(X) VARCHAR2(x)	Opção usual  Em caso excepcional como número de processo, placa de automóvel, número de chassi, ou demais numerais expressos com caracteres alfanuméricos
<b>IN</b>	NUMBER(1)	Valores permitidos 0 para Não e 1 para Sim
<b>DT</b>	DATE NUMBER(X)	Opção usual  Quando tipo de dado for mais adequado ao modelo do que DATE, como quando se refere a uma data parcial como uma competência (aaaamm)
<b>HR</b>	DATE NUMBER(X)	Opção usual  Quando o tipo de dado for mais adequado ao modelo do que DATE, como quando se refere a uma hora parcial como "mmss"



**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

Classe	Tipo de Dado	Observações
<b>NM</b>	VARCHAR2(X)	Especificar o comprimento máximo do atributo
<b>TE</b>	VARCHAR2(X) CLOB STRING	Especificar o comprimento máximo do atributo  Utilizado a nível lógico quando o tamanho do texto a ser inserido é maior que 65535 bytes ou não possui limite conhecido  Utilizado a nível físico quando o tamanho do texto a ser inserido é maior que 65535 bytes ou não possui limite conhecido
<b>VL</b>	NUMBER(X)	Especificar o valor máximo do atributo
<b>QT</b>	NUMBER(X)	Especificar o valor máximo do atributo
<b>TX</b>	NUMBER(X)	Especificar o valor máximo do atributo
<b>SG</b>	VARCHAR2(X)	Especificar o comprimento máximo do atributo
<b>IM</b>	BLOB	
<b>VD</b>	BLOB	
<b>SO</b>	BLOB	
<b>DO</b>	BLOB	

**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS****5.9 Tipos de dados para SGBD PostgreSQL**

Classe	Tipo de Dado	Observações
<b>ID</b>	NUMERIC(X)	Especificar o tamanho do atributo de acordo com o volume máximo da tabela, acrescentando-se em média dois dígitos conforme a expectativa de crescimento
<b>CS</b>	NUMERIC(X) VARCHAR(X)	Especificar o valor máximo do atributo
<b>CE</b>	NUMERIC(X) VARCHAR(X)	Especificar o valor máximo do atributo
<b>NU</b>	NUMERIC(X) VARCHAR(X)	Especificar o valor máximo do atributo e decimais, se houver  Em caso excepcional como número de placa, chassi, processo, ou demais numerais expressos com caracteres alfanuméricos
<b>IN</b>	NUMERIC(1)	Valores permitidos 0 para Não e 1 para Sim
<b>DT</b>	DATE NUMERIC(X)	Armazena somente a Data  Quando tipo de dado for mais adequado ao modelo do que DATE, como quando se refere a uma data parcial como uma competência (aaaamm)
<b>HR</b>	TIME NUMERIC(X)	Armazena somente a hora do dia  Quando o tipo de dado for mais adequado ao modelo do que TIME, como quando se refere a uma hora parcial como "mms"

**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

<b>Classe</b>	<b>Tipo de Dado</b>	<b>Observações</b>
<b>NM</b>	VARCHAR(X)	Especificar o comprimento máximo do atributo
<b>TE</b>	VARCHAR(X) TEXT	Especificar o comprimento máximo do atributo Utilizado quando o tamanho do texto a ser inserido não possui limite de caracteres conhecidos
<b>VL</b>	NUMERIC(X)	Especificar o valor máximo do atributo
<b>QT</b>	NUMERIC(X)	Especificar o valor máximo do atributo
<b>TX</b>	NUMERIC(X)	Especificar o valor máximo do atributo
<b>SG</b>	VARCHAR(X)	Especificar o comprimento máximo do atributo
<b>IM</b>	BLOB	Será convertido no modelo físico para BYTEA
<b>VD</b>	BLOB	Será convertido no modelo físico para BYTEA
<b>SO</b>	BLOB	Será convertido no modelo físico para BYTEA
<b>DO</b>	BLOB	Será convertido no modelo físico para BYTEA





## PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS

5.10 Tipos de dados para *HADOOP/Hive*

Classe	Tipo de Dado	Observações
<b>ID</b>	DECIMAL(X)	Especificar o tamanho do atributo de acordo com o volume máximo da tabela. Admite-se tamanhos de até três vezes o volume máximo ou dois dígitos.
<b>CS</b>	DECIMAL(X) VARCHAR(X)	Especificar o valor máximo do atributo
<b>CE</b>	DECIMAL(X) VARCHAR(X)	Especificar o valor máximo do atributo
<b>NU</b>	DECIMAL(X) VARCHAR(X)	Opção usual  Em caso excepcional como número de processo, placa de automóvel, número de chassi, ou demais numerais expressos com caracteres alfanuméricos
<b>IN</b>	DECIMAL(1)	Valores permitidos 0 para Não e 1 para Sim
<b>DT</b>	DATE DECIMAL(X)	Opção usual  Quando tipo de dado for mais adequado ao modelo do que DATE, como quando se refere a uma data parcial como uma competência (aaaamm)
<b>HR</b>	TIMESTAMP DECIMAL(X)	Opção usual  Quando o tipo de dado for mais adequado ao modelo do que TIMESTAMP, como quando se refere a uma hora parcial como “mmss”



**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

Classe	Tipo de Dado	Observações
<b>NM</b>	VARCHAR(X)	Especificar o comprimento máximo do atributo
<b>TE</b>	VARCHAR(X) CLOB	Especificar o comprimento máximo do atributo Utilizado quando o tamanho do texto a ser inserido é maior que 4000 bytes ou não possui limite
<b>VL</b>	DECIMAL(X)	Especificar o valor máximo do atributo
<b>QT</b>	DECIMAL(X)	Especificar o valor máximo do atributo
<b>TX</b>	DECIMAL(X)	Especificar o valor máximo do atributo
<b>SG</b>	VARCHAR(X)	Especificar o comprimento máximo do atributo
<b>IM</b>	BINARY	
<b>VD</b>	BINARY	
<b>SO</b>	BINARY	
<b>DO</b>	BINARY	

**5.11 Nomenclatura para demais elementos nos modelos de dados**

Elemento	Nome padronizado	Exemplo
Chave Primária	{abreviação da tabela}_PK	ACA_PK
Índice de Chave Primária	{nome da chave primária}	ACA_PK



**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

Chave Estrangeira	{abreviação da tabela}_{abreviação da tabela "pai"}_FK	ACA_CEPS_FK
Índice de Chave Estrangeira	{nome da chave estrangeira}_I	ACA_CEPS_FK_I
Índice	{abreviação da tabela}_{nome significativo}_I	ACA_ENDERECO_I
Chave Única	{abreviação da tabela}_{nome significativo}_UK	MUNI_NOME_UK
Regra de Verificação	{abreviação da tabela}_{nome significativo}_CK	FUN_SALARIO_CK
Domínio	DM_{nome significativo}	DM_ESTADO_CIVIL
Sequence	SQ_{nome da tabela associada no plural}	SQ_ACOES
Visão	VI_{nome significativo}	VI_FUNCIONARIOS_ATIVOS
Visão Materializada	MV_{nome significativo}	MV_MUNICIPIOS
Procedure	PR_{nome significativo}	PR_ATUALIZA_HISTORICO
Trigger	TG_{nome significativo}	TG_ATUALIZA_LOG
Function	FC_{nome significativo}	FC_IDENTIFICA_USUARIO
Package	PKG_{nome significativo}	PKG_CARGA_HISTORICO

**6 PROCESSO DE INTEGRAÇÃO DE MODELOS DE DADOS**

As bases de dados de um cliente devem ser entendidas como peças de um modelo de dados

**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

maior, o qual busca representar o negócio do cliente, independente da tecnologia de armazenamento adotada. A construção de modelos de dados ou processo de modelagem de dados, como também pode ser chamado, privilegiando a construção de modelos de dados altamente integrados apresenta os seguintes benefícios:

- (i) Entrada de dados em pontos únicos, evitando a necessidade de sincronização cruzada entre modelos de dados;
- (ii) A redundância de dados é evitada, economizando espaço de armazenamento;
- (iii) Os modelos de dados são mantidos coesos, coerentes e responsáveis por um subconjunto dos dados que fazem parte do negócio do cliente.

Desta forma, pregasse que a integração dos modelos de dados deve ser um dos principais objetivos no processo de construção de cada modelo de dado. De fato, a análise dos dados que comporão um novo modelo de dados e como ele pode ser incorporado ao modelo de dados do cliente, deve ser uma das primeiras atividades do processo de modelagem de dados. Dentre as principais atividades quanto ao processo de integração na construção de modelos de dados, destacam-se:

(1) Novos atributos e entidades devem ser avaliadas quanto a que assunto ou negócio do cliente estes pertencem e se estes já são tratados em alguns dos modelos de dados já existentes.

(2) Uma vez entendido que de fato representam um conceito não abordado em algum dos modelos de dados já existentes, as novas entidades e atributos devem ser avaliados quanto ao seu uso pelos demais modelos de dados existentes do cliente. Caso possam ser utilizados por vários dos demais modelos, estes devem ser incluídos em um dos modelos de dados compartilhados do cliente ou até cogitada sua incorporação ao Modelo de Dados Corporativo caso sejam de interesse de outros clientes.

Uma exceção a ser observada quanto a evitar a redundância de dados, diz respeito às aplicações onde a comunicação entre as fontes de dados seja um gargalo tecnológico e o desempenho seja um requisito crítico. Nesta situação se considera a clonagem dos dados como uma técnica adequada. No entanto, este artifício deve ser utilizado com cautela, não permitindo que se crie outro ponto de entrada de informações para os dados clonados e jamais permitindo que estes sofram atualizações.

### **6.1 Processo de Integração nas Aplicações Analíticas.**

Aplicações analíticas usualmente precisam lidar com grandes volumes de dados, desta forma a exceção quanto a evitar a redundância de dados explanada na seção anterior é justamente uma das técnicas empregadas como meio de garantir a viabilidade desta categoria de aplicações.

Nas aplicações analíticas a consulta às fontes de dados de forma online inviabilizaria as aplicações. A comunicação entre aplicações analíticas e suas fontes de dados, devido ao

**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

desempenho na comunicação de grandes volumes de dados, é justamente um gargalo tecnológico, onde a técnica de clonagem se torna necessária.

Desta forma, quanto ao processo de integração quando referenciamos a aplicações analíticas, temos um terceiro passo:

(3) Analisar as fontes de dados a serem utilizadas na construção do modelo de dados e cloná-las semanticamente idênticas ao dado presente na origem e apenas a partir destas entidades clonadas se recomenda implementar as transformações necessárias.

Por semanticamente idênticas ao dado presente na origem, quer se dizer que traduções necessárias na migração do dado entre tecnologias de armazenamento são toleradas.

Com relação às técnicas de clonagem disponíveis, na Dataprev estas tem evoluído com o tempo e conforme as tecnologias de armazenamento foram sendo incorporadas. Cada tecnologia de armazenamento possui suas indicações quanto à forma mais adequada de implementar a clonagem, mas independente da técnica de clonagem adotada, os principais pontos de atenção são:

(i) Não permitir atualizações nos dados clonados que necessitem ser retornados às fontes de dados de origem;

(ii) Não permitir operações de transformação entre a fonte de dados de origem e o dado clonado, preservando a semântica original do dado.

De fato, impedir que haja atualizações nos dados clonados evita a necessidade de implementação de complexas sincronizações cruzadas. Por sua vez, preservar a semântica original do dado permite que este seja reusado por vários processos dentro de uma mesma aplicação ou por várias aplicações, dentre de uma mesma tecnologia de armazenamento ou servidor. No mais, também reduz o ônus das extrações nas fontes de dados de origem. As alterações necessárias às aplicações nas entidades clonadas devem ser tratadas em uma segunda camada de dados específica, usando estas entidades clonadas como fonte de dados.

A exceção à regra é quando não há possibilidade ou interesse no reuso do dado da fonte de origem por outras aplicações, neste caso não haverá ônus desnecessário de outros processos de extração e carga de dados e a transformação pode ser tolerada que aconteça em conjunto com o processo de extração e carga de dados.

**7 ARQUITETURA DE DADOS PARA APLICAÇÕES ANALÍTICAS**

A arquitetura de Projetos de Dados para aplicações analíticas na Dataprev pode ser entendida como uma arquitetura multicamada e em parte este aspecto da arquitetura se deve aos requisitos de integração mencionados na seção anterior. Esta arquitetura pode ser entendida como adequada para todas as tecnologias de armazenamento utilizadas na Dataprev, sejam estas baseadas em SGBDs Objeto/Relacionais ou mesmo tecnologias de armazenamento “Não SQL”, conforme será discutido a seguir.

**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

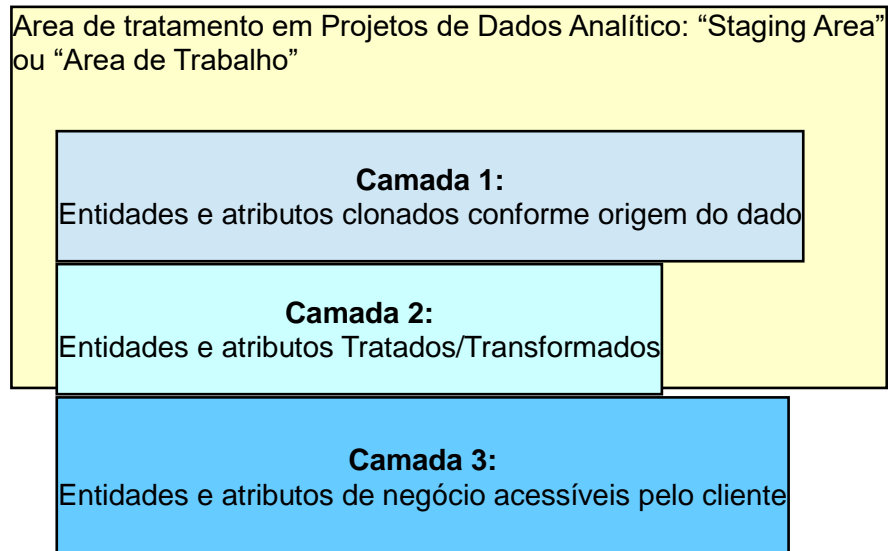
A primeira camada diz respeito aos dados clonados a partir das fontes de dados da origem e é persistida em uma área ou local usualmente chamado de “Staging Area” (SA) ou “Área de Trabalho” (AT), quando nos referimos a termos de aplicações analíticas. Nesta primeira camada se entende que devem haver apenas as estruturas de dados clonadas semanticamente conforme presentes na origem. Esta camada segue a regra mencionada como forma de propiciar a integração através de seu reuso, pelos motivos explanados na seção anterior. Como também mencionado anteriormente a exceção à regra se dá quando não há possibilidade ou interesse no reuso do dado.

A segunda camada representa as entidades e atributos resultantes ou utilizadas em transformações necessárias à aplicação analítica. Esta deve utilizar a primeira camada como fonte de dados e assim construir as entidades e atributos específicos que sejam necessários à aplicação analítica em questão sendo construída. Esta pode ser entendida ainda como sendo persistida na área chamada de SA, pois seu objetivo ainda é preparar os dados para a próxima camada a qual conterá os dados disponibilizados ao cliente.

Por sua vez, a terceira camada tem por objetivo representar as entidades do negócio objetivo da aplicação analítica propriamente dita. O local onde a camada pode ser persistida se confunde com os termos utilizados para definir o próprio tipo da aplicação analítica, uma vez que as entidades persistidas nesta são o principal objetivo desta. Desta forma, podemos chamá-la de diferentes formas conforme o tipo de aplicação analítica sendo abordada, mas as denominações mais comuns são: DM, DW, ODS ou Data Lake. Esta representa o objetivo final da aplicação analítica que é dispor os dados ao cliente, seja através de uma ferramenta, aplicação ou diretamente dependendo do tipo da aplicação em questão, no formato adequado para a geração de informações analíticas. Assim, ela representa de fato a aplicação analítica final, conforme exemplificado na figura 1.

Importante ressaltar que podem haver mais camadas conforme as necessidades de transformação do dado ou ainda visões deste que o usuário necessite. No mais, embora conceitualmente apartadas, a forma como as entidades de cada uma destas camadas será fisicamente implementada pode variar conforme a tecnologia de armazenamento adotada, podendo estar alocadas em unidades de armazenamento separadas ou até fisicamente implementadas no mesmo nível hierárquico/esquema.

**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**



*Figura 1: Arquitetura de Dados multicamada para Aplicações Analíticas x Conceitos de Aplicações Analíticas.*



**ADMINISTRAÇÃO DE DADOS**

**NORMA**

**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

**8 VIGÊNCIA**

Esta Norma entra em vigor a partir desta data e revoga a N/AD/001/04.

**DIOGO COSTA MARTINS PIZANESCHI**  
Responsável pela elaboração  
**SUPERINTENDENTE**

**JOSE IVANILDO DIAS JUNIOR**  
Responsável pela elaboração  
**CONSULTOR JURÍDICO**

**ANTÔNIO RICARDO DE OLIVEIRA JUNQUEIRA**  
Responsável pela aprovação  
**DIRETOR**





**PADRÕES PARA CONSTRUÇÃO DE MODELOS DE DADOS**

**ANEXO 1**

**REFERÊNCIAS**

BARBIERI, Carlos. **BI – Business Intelligence: Modelagem e Tecnologia**. 1ª Edição. Rio de Janeiro: Axcel Books, 2001

COUGO, Paulo. **Modelagem Conceitual e Projeto de Banco de Dados**. 1ª Edição. Rio de Janeiro: Campus, 1997.

DATE, C J. **Introdução a Sistemas de Bancos de Dados**. 8ª Edição. Rio de Janeiro: Elsevier, 2003.

KORTH, Henry F.; SILBERSCHATZ, Abraham; SUDARSHAN, S. **Sistema de Banco de Dados**. 3ª Edição. São Paulo: Makron Books, 1999.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistema de Banco de Dados**. 4ª Edição. São Paulo: Addison Wesley, 2005.

REGO, Bergson Lopes. **Gestão e governança de dados: promovendo os dados como ativo de valor nas empresas**. 1ª Edição. Rio de Janeiro: Brasport, 2013.

TEOREY, Tobey J. **Projeto e modelagem de banco de dados**. 2ª Edição. Rio de Janeiro: Elsevier, 2014.

MILANI, André. **PostgreSQL: Guia do programador**. 1ª Edição. São Paulo: Novatec Editora, 2008.

INMON, W.H.; HACKARTHORN, R. D. **Como usar o Data Warehouse**. 2ª edição. Rio de Janeiro: IBPI Press, 1997.

INMON, W.H.; IMHOFF, Claudia; BATTAS, Greg. **Building the Operational Data Store**. 1ª Edição. New York: John Wiley & Sons, Inc., 1996.

INMON, W.H.; IMHOFF, Claudia; SOUSA Ryan. **Corporate Information Factory**. 2ª edição. New York: Wiley, 2001.



**Assinado digitalmente por:**

Antonio Ricardo de Oliveira Junqueira (Aprovador)

Jose Ivanildo Dias Junior (Chancelador)

Diogo Costa Martins Pizaneschi (Elaborador)