

Criação do Dicionário de Dados

1. Introdução ao Dicionário de Dados

1.1 Definição de Dicionário de Dados

O **dicionário de dados** é um repositório centralizado de informações sobre os dados contidos em um sistema ou banco de dados. Ele documenta as descrições dos elementos de dados, suas características, suas relações e as regras de negócios associadas.

Objetivos do Dicionário de Dados:

- **Organização e padronização dos dados:** Fornecer uma visão clara e padronizada dos dados utilizados no sistema.
- **Facilitar a comunicação entre equipes:** Servir como uma referência comum para desenvolvedores, analistas de negócios e outras partes interessadas.
- **Melhoria na qualidade dos dados:** Garantir a consistência, integridade e precisão dos dados no sistema.
- **Documentação e governança:** Manter um registro formal das definições e metadados associados aos elementos de dados, promovendo boas práticas de governança.

1.2 Importância do Dicionário de Dados

- **Facilita a compreensão:** Fornece uma visão clara de como os dados são organizados, usados e mantidos.
 - **Promove a consistência:** Garante que todos os envolvidos no projeto compreendam os dados da mesma forma, eliminando ambiguidades.
 - **Suporte à governança de dados:** Ajuda a garantir que as práticas de segurança e conformidade sejam seguidas.
 - **Facilita a manutenção do sistema:** Permite a rápida localização de informações relevantes sobre os dados para atualizações e correções no sistema.
-

2. Componentes do Dicionário de Dados

Um dicionário de dados contém uma série de elementos que descrevem as propriedades dos dados. Vamos explorar os principais componentes que compõem um dicionário de dados.

2.1 Elementos de Dados

Os **elementos de dados** são as unidades individuais descritas no dicionário de dados. Cada elemento representa um campo ou atributo no banco de dados.

Principais Informações dos Elementos de Dados:

1. **Nome do Campo:** Identificação única do elemento de dado. Deve ser claro e descritivo.
 - Exemplo: `nome_cliente`, `id_produto`.
2. **Descrição do Campo:** Definição clara e detalhada do campo, explicando o que ele representa.
 - Exemplo: `id_produto`: "Identificador único para cada produto no sistema."
3. **Tipo de Dado:** Define o formato do dado armazenado, como:
 - **Texto (VARCHAR, CHAR):** Utilizado para armazenar palavras, frases ou identificadores.
 - **Números (INT, FLOAT, DECIMAL):** Utilizado para armazenar valores numéricos inteiros ou com casas decimais.
 - **Datas (DATE, DATETIME, TIMESTAMP):** Usado para armazenar datas e horas.
 - **Booleanos (BOOLEAN):** Armazena valores Verdadeiro (`TRUE`) ou Falso (`FALSE`).
4. **Tamanho ou Precisão:** Define o comprimento máximo do dado ou a precisão numérica.
 - Exemplo: `VARCHAR(50)`, `DECIMAL(10,2)`.
5. **Domínio de Valores:** Define os valores possíveis para aquele campo, se aplicável.
 - Exemplo: Um campo booleano pode ter os valores `TRUE` ou `FALSE`.
6. **Chave Primária/Chave Estrangeira:** Indica se o campo é uma chave primária (única para identificar o registro) ou uma chave estrangeira (relaciona-se com outro dado em uma tabela diferente).
 - Exemplo: `id_produto` é chave primária na tabela **Produtos** e chave estrangeira na tabela **Pedidos**.
7. **Restrições de Integridade:** Regras que garantem a qualidade e consistência dos dados. Exemplos:
 - **NOT NULL:** O campo não pode ser vazio.
 - **UNIQUE:** O valor do campo deve ser único.
 - **DEFAULT:** Um valor padrão é atribuído quando nenhum dado é fornecido.
8. **Unidade de Medida:** Quando aplicável, especifica a unidade associada ao valor armazenado.
 - Exemplo: Campo `peso_produto`, unidade: `kg`.

2.2 Tabelas e Relacionamentos

Além dos campos individuais, o dicionário de dados também documenta a estrutura das **tabelas** e os **relacionamentos** entre elas.

1. **Nome da Tabela:** Nome que identifica a entidade armazenada.
 - Exemplo: Tabela **Clientes**, Tabela **Produtos**.
2. **Chaves Primárias e Estrangeiras:** As chaves primárias identificam registros únicos em uma tabela, enquanto as chaves estrangeiras indicam relações com outras tabelas.
 - Exemplo: Na tabela **Pedidos**, **id_cliente** pode ser uma chave estrangeira que referencia **id_cliente** na tabela **Clientes**.
3. **Relacionamentos:** Documentação de como as tabelas se relacionam, como:
 - **1:1:** Um para um, quando cada registro em uma tabela corresponde a um único registro em outra tabela.
 - **1**
: Um para muitos, quando um registro em uma tabela pode se relacionar com vários registros em outra tabela.
 - **N**
: Muitos para muitos, quando múltiplos registros em uma tabela podem se relacionar com múltiplos registros em outra.

2.3 Regras de Negócio

As **regras de negócio** são condições ou restrições que definem como os dados devem ser manipulados dentro do sistema, garantindo que os requisitos do negócio sejam atendidos.

Exemplos de Regras de Negócio:

- **Descontos:** "Clientes com mais de 10 pedidos recebem um desconto de 10%."
 - **Estoque Mínimo:** "Se o estoque de um produto cair abaixo de 20 unidades, um alerta deve ser enviado."
 - **Validação de Dados:** "Números de CPF devem ter exatamente 11 dígitos."
-

3. Etapas para a Criação do Dicionário de Dados

3.1 Identificação de Elementos de Dados

A primeira etapa para a criação de um dicionário de dados é identificar todos os elementos de dados que serão usados no sistema. Isso pode ser feito durante a fase de **levantamento de requisitos e modelagem conceitual**.

Processo de Identificação:

1. Revisar documentos de requisitos do sistema.
2. Realizar entrevistas com usuários e stakeholders.
3. Analisar sistemas existentes para identificar dados importantes.
4. Identificar entidades e atributos necessários para a operação do sistema.

3.2 Documentação dos Elementos de Dados

Após identificar os elementos, é hora de documentá-los detalhadamente no dicionário de dados. A documentação deve ser clara e concisa, fornecendo todas as informações relevantes para cada elemento.

Dicas para Documentação:

- Use uma **nomenclatura padronizada** para nomear campos e tabelas, evitando ambiguidades.
- Inclua todos os detalhes relevantes, como **tipo de dado, domínio de valores, restrições e relacionamentos**.
- Organize os elementos de dados em categorias lógicas para facilitar a navegação e a consulta.

3.3 Validação do Dicionário de Dados

Uma vez documentados os dados, é essencial validar o dicionário com as partes interessadas para garantir que ele reflita corretamente as necessidades do sistema.

Etapas de Validação:

1. Revisar o dicionário de dados com a equipe de desenvolvimento e os usuários.
 2. Verificar se todas as regras de negócio foram corretamente refletidas.
 3. Testar cenários com os dados para garantir consistência e precisão.
-

4. Boas Práticas na Criação do Dicionário de Dados

4.1 Manter o Dicionário Atualizado

- O dicionário de dados deve ser tratado como um **documento vivo**, ou seja, ele deve ser atualizado conforme o sistema evolui e novas informações ou dados são adicionados.
- Sempre que uma nova tabela ou campo for adicionado ao sistema, o dicionário de dados deve ser revisado e ajustado.

4.2 Documentação Clara e Precisa

- Certifique-se de que os nomes dos elementos de dados sejam claros e autodescritivos. O objetivo é que qualquer pessoa que consulte o dicionário de dados entenda rapidamente o que cada campo representa.
- Evite abreviações e jargões técnicos que possam gerar confusão.

4.3 Automação no Processo de Documentação

- Sempre que possível, utilize ferramentas de desenvolvimento que ofereçam **automação para a geração de dicionários de dados**. Muitas ferramentas de modelagem de banco de dados, como **MySQL Workbench** e **Oracle SQL Developer**, permitem a exportação automática do esquema de banco de dados diretamente para um dicionário de dados. Isso reduz o risco de erros manuais e garante que o dicionário esteja sempre em sincronia com o banco de dados real. Além disso, essas ferramentas permitem a exportação em diferentes formatos, como PDF, HTML ou Excel, facilitando a distribuição e o compartilhamento entre as equipes.

Ferramentas Comuns para Automação:

- **MySQL Workbench**: Gera o modelo ER (Entidade-Relacionamento) e permite a exportação automática de um dicionário de dados a partir do modelo.
- **Oracle SQL Developer**: Permite criar e visualizar o dicionário de dados de forma automatizada, com opções de exportação detalhadas.
- **SQL Server Management Studio (SSMS)**: Gera um esquema completo do banco de dados com documentação automática dos campos e tipos de dados.
- **DBDesigner**: Uma ferramenta visual que permite a criação de modelos de banco de dados e exportação de dicionários de dados de forma simples.

4.4 Padronização de Nomenclaturas

- Ao criar um dicionário de dados, a **padronização** é essencial para manter a clareza e a consistência dos nomes dos campos e tabelas. O uso de uma nomenclatura padronizada facilita a navegação e compreensão dos dados, além de reduzir ambiguidades.

Regras de Padronização:

- **Nomeação Descritiva:** Nomes de campos e tabelas devem descrever claramente o conteúdo ou propósito do dado.
 - Exemplo: `nome_cliente`, `data_nascimento`, `quantidade_em_estoque`.
- **Uso de Idioma Consistente:** Escolha um idioma para todos os elementos (português ou inglês, por exemplo) e mantenha-se consistente.
- **Convenções de Capitalização:** Defina uma convenção de capitalização, como usar *camelCase* ou *snake_case*.
 - **camelCase:** `nomeCliente`, `dataNascimento`.
 - **snake_case:** `nome_cliente`, `data_nascimento`.

Benefícios da Padronização:

- Facilita a **comunicação entre equipes**, pois todos seguem o mesmo padrão.
 - Reduz erros durante o **desenvolvimento e manutenção** do banco de dados.
 - Melhora a **navegabilidade** em bancos de dados grandes e complexos.
-

4.5 Uso de Comentários e Anotações

- O uso de **comentários** nos campos e tabelas pode fornecer informações adicionais que não estão necessariamente cobertas pelas descrições dos campos. Isso é especialmente útil em sistemas complexos, onde um campo pode ter uma função que não é óbvia à primeira vista.

Dicas para o Uso de Comentários:

- Inclua **justificativas** para regras de negócios implementadas em determinados campos.
 - Exemplo: “Este campo é utilizado para armazenar a quantidade mínima permitida em estoque para ativar pedidos automáticos.”
- Utilize os comentários para **explicar exceções ou particularidades** no uso dos dados.
 - Exemplo: “Este campo é opcional para clientes internacionais.”
- Insira anotações sobre **modificações feitas nos dados**, indicando a versão ou data de alterações importantes.
 - Exemplo: “Campo adicionado na versão 2.1 para permitir a gestão de múltiplos endereços por cliente.”

Ferramentas de Documentação que Suportam Comentários:

- **SQL Server:** Suporta a adição de comentários diretamente no esquema do banco de dados.
- **MySQL:** Permite a adição de comentários em colunas e tabelas.
- **PostgreSQL:** Oferece uma ampla funcionalidade de comentários para tabelas, colunas, índices, e outros elementos.

4.6 Governança de Dados e Controle de Acessos

- O dicionário de dados também pode ser utilizado para **governança de dados**, garantindo que as políticas de segurança e integridade sejam aplicadas corretamente. A governança de dados está diretamente relacionada à forma como os dados são gerenciados, acessados e protegidos.

Considerações para Governança:

- **Controle de Acesso:** Definir quem pode visualizar, modificar ou excluir determinados dados no sistema. Isso pode ser documentado no dicionário de dados, onde cada campo ou tabela tem um nível de permissão atribuído.
 - Exemplo: Apenas administradores têm permissão para modificar dados financeiros.
- **Políticas de Retenção de Dados:** Definir o período durante o qual os dados devem ser armazenados antes de serem arquivados ou excluídos.
 - Exemplo: “Os registros de clientes inativos devem ser arquivados após 5 anos.”
- **Políticas de Backup:** Estabelecer regras sobre quando e como os dados devem ser copiados para fins de segurança.
 - Exemplo: “Backups diários são realizados para todas as tabelas transacionais.”

4.7 Revisão e Atualização Periódica

- Como o dicionário de dados é um documento vivo, ele deve ser revisado periodicamente para garantir que todas as modificações no sistema ou no banco de dados sejam refletidas.

Passos para Manutenção do Dicionário de Dados:

1. **Revisão Programada:** Definir intervalos regulares (ex.: trimestrais ou semestrais) para revisar o dicionário de dados e verificar sua conformidade com o banco de dados atual.
2. **Validação com as Equipes:** Envolver os desenvolvedores e analistas de negócios para validar se as descrições, tipos de dados e regras de negócio continuam corretas.
3. **Controle de Versões:** Manter um registro das versões do dicionário de dados, com uma lista de mudanças implementadas a cada atualização.

Ferramentas de Versionamento:

- **Git** ou outros sistemas de controle de versão podem ser utilizados para rastrear as mudanças no dicionário de dados ao longo do tempo.
- Utilizar ferramentas específicas de versionamento de bancos de dados como **Liquibase** ou **Flyway**.

Conclusão: A Importância do Dicionário de Dados na Governança de Informações

- O dicionário de dados é muito mais do que um simples repositório de descrições de campos e tabelas. Ele desempenha um papel crucial na governança de dados, comunicação entre equipes, padronização de nomenclaturas, e manutenção da segurança e integridade dos dados. Manter o dicionário atualizado, padronizado e com documentação clara é uma das melhores práticas para garantir o sucesso e a longevidade de qualquer sistema de banco de dados.