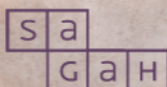


ADMINISTRAÇÃO DE BANCO DE DADOS

Claudia Abreu Paes



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS



Dicionário de dados

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Descrever o que é um dicionário de dados.
- Explicar as funcionalidades do dicionário de dados.
- Prever a criação de um dicionário de dados.

Introdução

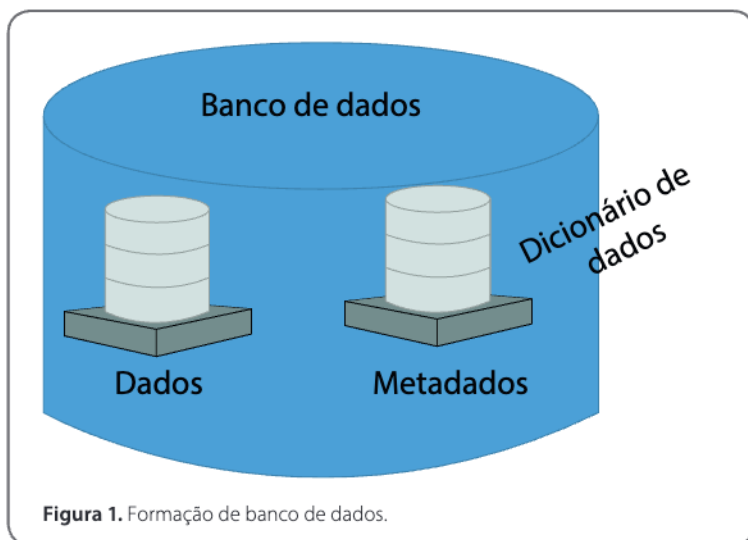
A documentação dos sistemas que são desenvolvidos é fundamental para a garantia de continuidade. Não é diferente quando se disponibiliza um banco de dados, pois sua estrutura incorpora conceitos, características próprias e regras que devem estar disponíveis às equipes de desenvolvimento. Nesse sentido, os dicionários de dados representam a documentação da estrutura do banco de dados. Tal estrutura é definida desde o momento de modelagem conceitual dos dados e utilizada sempre que se deseja conhecer as características do banco de dados.

Neste capítulo, você vai estudar conceitos básicos de dicionário de dados aplicados a banco de dados. Você vai ver a funcionalidade e a importância do dicionário de dados, bem como os procedimentos necessários à sua criação.

O que é o dicionário de dados?

Segundo Silberschatz, Korth e Sundarshan (2012), um dicionário de dados é um repositório que armazena metadados (informação sobre dados) a respeito da estrutura do banco de dados. Bittencourt (2004) define as colunas que compõem cada tabela, as restrições de integridade, as informações de segurança e outros elementos estruturais (como visualizações e domínios definidos pelo usuário) como tipos de informações que formam a estrutura de um dicionário de dados.

Dessa forma, o banco de dados armazena os dados e também as suas estruturas (denominação, características de tipo e tamanho e regras de integridade), denominadas metadados. É o que você pode ver na Figura 1.

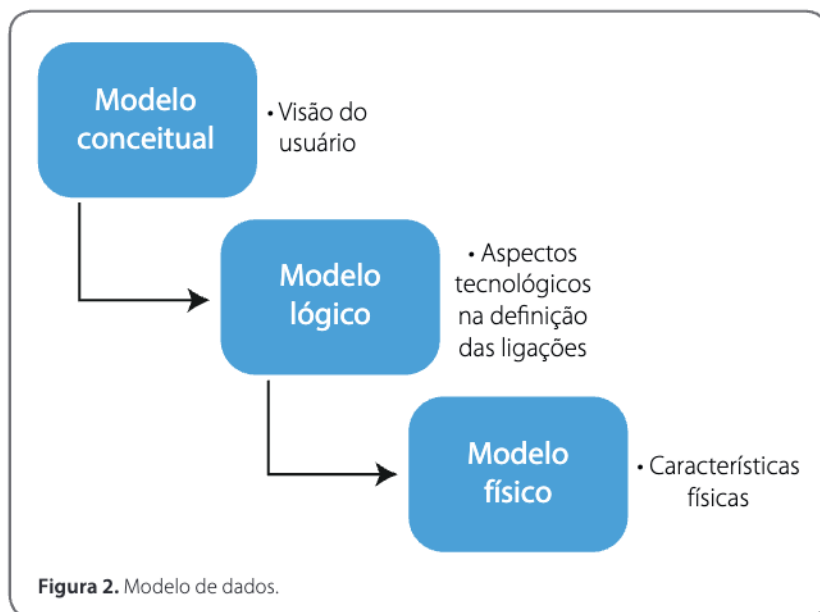


Segundo o *Manual de Referência MySQL* (MYSQL..., c2019), o dicionário de dados possui alguns benefícios:

- armazenamento uniforme dos dados do dicionário de dados em um único *schema*;
- remoção de armazenamento de metadados baseado em arquivo;
- armazenamento transacional e seguro de dados do dicionário;
- armazenamento em cache uniforme e centralizado para objetos de dicionário;
- declaração de definição de dados atômicos, denominado DDL atômico.

A construção do dicionário de dados tem início a partir do modelo de dados. Segundo Silberschatz, Korth e Sundarshan (2012), para que se possa usar um sistema, ele deve ser eficiente na recuperação das informações. Essa eficiência está relacionada à forma pela qual foram projetadas as complexas estruturas de representação dos dados no banco de dados. Daí a importância da criação do modelo de dados.

Segundo Heuser (2009), um modelo de dados é uma descrição dos tipos de informações que estão armazenadas em um banco de dados. Vários níveis de abstração podem ser considerados para se modelar um banco de dados. Por meio do modelo de dados, o usuário pode entender a organização de um banco de dados. Os níveis de abstração considerados são: modelo conceitual, modelo lógico e modelo físico (Figura 2).



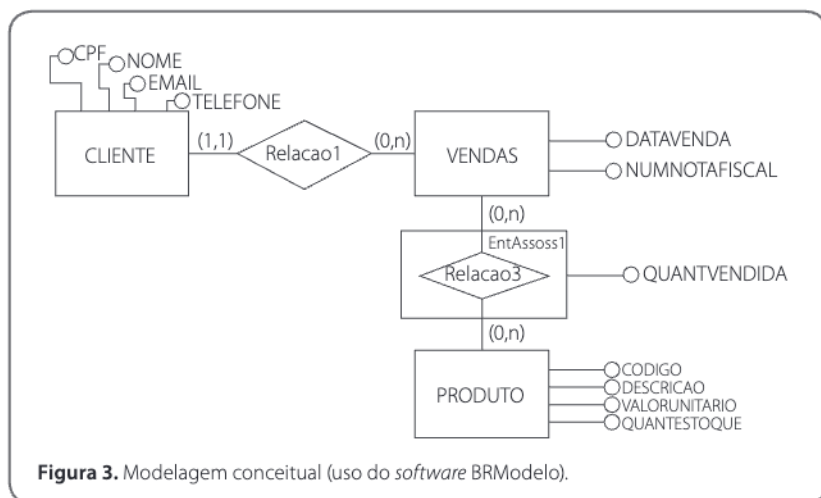
Modelagem conceitual

A modelagem conceitual é usada como representação de alto nível e considerada exclusivamente o ponto de vista do usuário criador dos dados. É nela que se usam os modelos de entidade e relacionamento para a representação dos conjuntos de elementos e seus relacionamentos. Nesse momento, os atributos são designados a cada entidade (conjunto de elementos).

Suponha que, em uma loja de calçados, o proprietário define que, para cada cliente, ele precisa anotar as informações CPF, nome, *e-mail* e telefone, para viabilizar o contato, caso seja necessário. O proprietário define ainda que, de um produto, é importante que se saiba seu código, sua descrição, seu valor unitário e sua quantidade em estoque. O proprietário complementa dizendo

que, em uma venda, são identificados a data e o número da nota fiscal gerada. Além disso, o cliente pode comprar mais de um produto, e um produto pode ser comprado por vários clientes.

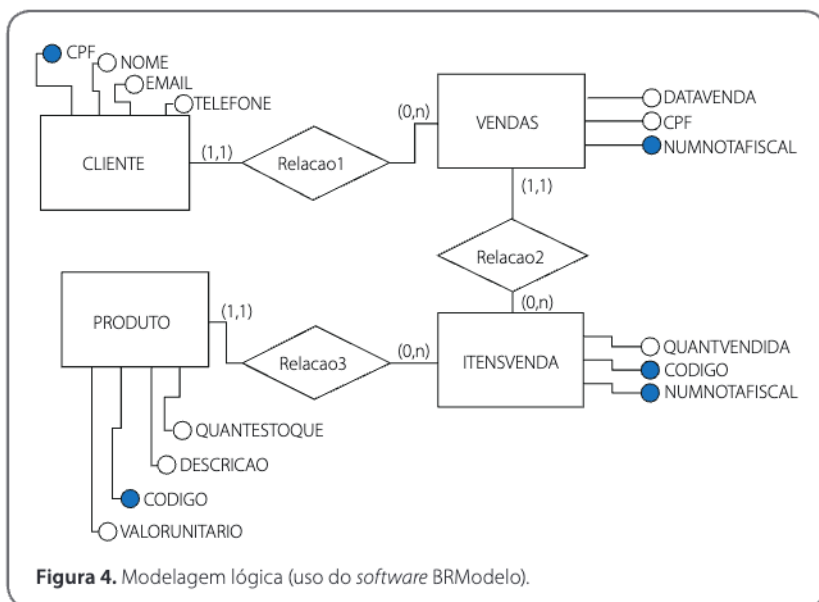
Para atender à especificação do proprietário da loja, propõe-se a modelagem conceitual dos dados e respectivos atributos, como você pode ver na Figura 3.



Modelagem lógica

A modelagem lógica agrega definições necessárias à implementação física, fazendo a derivação do modelo conceitual com a definição das chaves primárias e estrangeiras, representando o relacionamento no banco de dados físico.

Para entender melhor como funciona a modelagem lógica (Figura 4), considere a modelagem conceitual e a sua derivação. O primeiro passo da derivação é a escolha das chaves primárias. No segundo passo, criam-se novas entidades a partir das entidades associativas. Por fim, há o transporte das chaves primárias das entidades com participação *N* no relacionamento para as entidades com participação 1. Assim, se definem as chaves estrangeiras e a chave primária, no caso de entidades advindas de entidade associativa.



Modelagem física

A modelagem física define as características físicas de armazenamento dos dados (tipo, tamanho, domínio, regras). Para cada entidade, ocorre a criação de uma tabela e seus atributos. Considerando o modelo lógico (Figura 4), quatro tabelas serão formadas, como você pode ver a seguir.

Tabela CLIENTE:

	Tipo	Tamanho	Chave primária (PK)?	Chave estrangeira (FK)?	Domínio
CPF	Inteiro	11	X		
NOME	Alfanumérico	100			
EMAIL	Alfanumérico	80			
TELEFONE	Númérico	9			

Tabela VENDA:

	Tipo	Tamanho	Chave primária?	Chave estrangeira?	Domínio
NUMNO-TAFISCAL	Inteiro	10	X		
DATAVENDA	Data	8			
CPF	Inteiro	11		X	

Tabela ITENSVENDA:

Atributo	Tipo	Tamanho	Chave primária?	Chave estrangeira?	Domínio
NUMNO-TAFISCAL	Inteiro	10	X	X	
CODIGO	Inteiro	6	X	X	
QUANT-VENDIDA	Alfanu-mérico	80			

Tabela PRODUTO:

Atributo	Tipo	Tamanho	Chave primária?	Chave estrangeira?	Domínio
CODIGO	Inteiro	6	X		
DESCRICAO	Alfanu-mérico	50			<i>not null</i>
VALOR-UNITARIO	Decimal	(6,2)			
QUANT-ESTOQUE	Numérico	4			

A definição do dicionário de dados se inicia pelo modelo conceitual de dados, passa pela modelagem lógica e pela modelagem física e termina quando se criam as estruturas no banco de dados. Nesse momento, ainda podem ser acrescentadas informações referentes à segurança, ao índice e a outros recursos, como visão. Todo objeto criado no banco de dados terá suas informações armazenadas no dicionário de dados.

Portanto, o dicionário de dados mantém informações de vários níveis. Veja a seguir.

- **Gestão do banco de dados:** a gestão de um banco de dados passa pelo acompanhamento da utilização do espaço físico. Muitas vezes, é preciso expandir os espaços ou realizar procedimentos de compactação. Dessa forma, as informações referentes à alocação de espaço e à verificação de utilização são de extrema importância para o administrador do banco de dados (*DataBase Administrator* [DBA]).
- **Usuários do banco de dados:** os usuários são todos cadastrados com suas senhas de acesso para que o controle de acesso possa ser realizado. Além disso, as informações dos usuários são associadas aos objetos que criam, como índices, visões e as próprias tabelas.
- **Formação do banco de dados:** definição precisa sobre elementos de dados (tabelas) e seus atributos, como nome, características de conteúdo para armazenamento (tipo e tamanho) e restrições de integridade de domínio — por exemplo, quando uma coluna que se refere ao estado civil só pode receber uma lista de valores discretos: casado, solteiro, divorciado, união estável, viúvo.

Todas essas informações, seja de gestão, de usuários ou de formação, compõem o dicionário de dados. Dessa forma, algumas vantagens podem ser verificadas:

- minimização de erros na modelagem dos dados, na medida em que se tem fácil acesso à documentação das informações;
- composição da documentação do sistema, garantindo a sua continuidade;
- representação de um identificador dos objetos de banco de dados para a constante validação de recursos utilizados, pois toda operação só se realiza se as especificações das solicitações estiverem de acordo com as definições constantes no dicionário de dados.

Funcionalidade do dicionário de dados

O acesso aos dados é realizado por meio dos sistemas gerenciadores de banco de dados (SGBDs). Esses sistemas operam os bancos de dados, controlando todas as transações. Dessa forma, são responsáveis pela definição das estruturas, pelo armazenamento e pela recuperação de dados, pela segurança, pela consistência e pela concorrência. Os SGBDs servem para aplicações multiusuárias em que a coordenação entre muitos usuários é necessária.

O sistema de banco de dados é um conjunto de quatro componentes básicos (Figura 5): banco de dados, *hardware*, *software* e usuário.

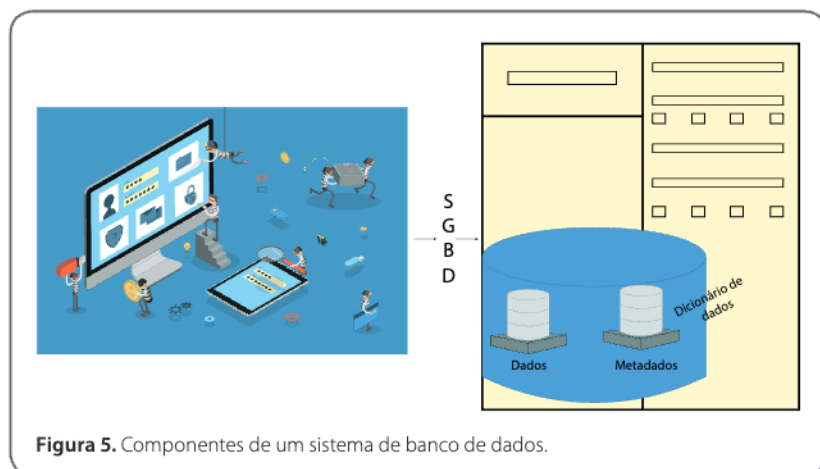


Figura 5. Componentes de um sistema de banco de dados.

Segundo Bittencourt (2004), quando um usuário tenta acessar dados de qualquer maneira, um SGBD relacional primeiro se dirige ao dicionário de dados para determinar se os elementos do banco de dados que o usuário solicitou fazem realmente parte do esquema. Além disso, o SGBD verifica se o usuário tem os direitos de acesso aos dados que ele está solicitando.

Quando um usuário tenta modificar ou excluir dados, o SGBD também vai para o dicionário de dados, dessa vez a fim de procurar restrições de integridade que podem ter sido colocadas na relação. Se os dados atendem às restrições, a modificação é permitida. Caso contrário, o SGBD retorna uma mensagem de erro e não faz a alteração.

Bittencourt (2004) afirma ainda que, como todo acesso ao banco de dados relacional é feito por meio do dicionário de dados, os SGBDs relacionais são baseados em um dicionário de dados. A formação de um dicionário de dados é representada por tabelas. Além disso, há uma tabela que documenta todas as tabelas do banco de dados.

Cada SGBD define o formato para armazenar o dicionário de dados. Também é comum as empresas adotarem diferentes convenções e terminologias com referência ao seu dicionário de dados. Assim, são determinados padrões de denominação para os atributos. Considere, por exemplo, a abreviação CD para o termo “código”. Quando for necessário fazer referência ao código de departamento, será preciso utilizar CDDepartamento. Já para o código de disciplina, será preciso utilizar CDDisciplina. Fato é que, dentro das especificações, quando alguém se depara com um nome de atributo iniciado por CD, já fica fácil de entender que é um código. Essa convenção traz alguns benefícios. Entre eles, considere os listados a seguir:

- melhor compreensão das definições;
- padronização da definição e uso dos itens de dados;
- criação de significados comuns;
- otimização da utilização das informações.

Há muitos SGBDs no mercado, como Oracle, SQLServer, MySQL e PostgreSQL. Aqui, a referência é o SGBD MySQL, por se tratar de um *software* livre utilizado em escala no mercado.

Estrutura do dicionário de dados

O dicionário de dados definido pelo SGBD MySQL realiza o armazenamento das informações em tabelas e visões no esquema denominado *information*. Além disso, as tabelas e visões armazenam informações de usuários, privilégios, objetos do banco de dados, regras de integridade e informações de auditoria.

As tabelas que compõem o dicionário de dados são utilizadas somente para leitura por meio do comando `SELECT`. Para cada tabela contida no dicionário, são apresentados os atributos relacionados. O resultado do comando apresenta um conjunto de linhas, correspondente aos elementos e colunas, identificando os elementos.

Para obter as informações do dicionário de dados, o usuário deve escolher a natureza da informação: colunas, restrição de integridade (*constraint*), tabelas, visões, privilégios, *tablespaces*, entre outras. Veja o exemplo a seguir, que considera as tabelas.

```
SELECT
```

```
TABLE_NAME, ENGINE, VERSION, ROW_FORMAT, TABLE_ROWS, AVG_ROW_LENGTH,  
DATA_LENGTH, MAX_DATA_LENGTH, INDEX_LENGTH, DATA_FREE, AUTO_INCREMENT,  
CREATE_TIME, UPDATE_TIME, CHECK_TIME, TABLE_COLLATION, CHECKSUM,  
CREATE_OPTIONS, TABLE_COMMENT
```

```
FROM INFORMATION_SCHEMA.TABLES
```

```
WHERE table_schema = 'db_name'
```

```
[AND table_name LIKE 'wild']
```

colunas tabela
TABLE

tabela

Note que as informações apresentadas são escritas em inglês pois esse é o idioma de construção do SGBD.



Link

A lista de tabelas utilizadas para armazenar o dicionário de dados no SGBD MySQL pode ser encontrada no *site* oficial de documentação:

<https://qr.go.page.link/6Aw9>

Ainda, é possível ter acesso às informações das colunas de uma tabela:

```
SELECT COLUMN_NAME, DATA_TYPE, IS_NULLABLE, COLUMN_DEFAULT  
FROM INFORMATION_SCHEMA.COLUMNS  
WHERE table_name = 'tbl_name'  
[AND table_schema = 'db_name']  
[AND column_name LIKE 'wild']
```

Todas as tabelas de dicionário de dados têm a mesma estrutura e devem obedecer às mesmas regras que as tabelas básicas: precisam ter chave primária única, não nula e integridade referencial entre elas próprias. Para a utilização, é necessário conhecer a composição de seus atributos.



Link

Para consultar a lista de atributos, acesse o *link* a seguir.

<https://qrgo.page.link/hAAc>

Criação do dicionário de dados

Nesta seção, você vai conhecer os procedimentos realizados para criar e visualizar o dicionário de dados.

Criação

Os SGBDs utilizam a linguagem de consulta estruturada (*Structured Query Language* [SQL]) para estabelecer a conexão com o banco de dados. A SQL surgiu na década de 1970. Ela se chamava Sequel e por fim foi definida como padrão de acesso a banco de dados pelo Instituto Nacional Americano de Padrões (ANSI). A SQL disponibiliza recursos de definição, manipulação, seleção, segurança e concorrência.

A criação do dicionário de dados se dá a partir das instruções realizadas durante a criação e/ou alteração das estruturas do banco de dados. Na SQL, o comando responsável pela criação das estruturas de dados é o CREATE. Como exemplo, considere o modelo físico a seguir.

Tabela DEPARTAMENTO:

Atri- buto	Tipo	Tama- nho	Chave primá- ria	Chave estran- geira	Domí- nio	Default
CDDE- PARTA- MENTO	Inteiro	5	X		<i>Not null</i>	
NMDE- PARTA- MENTO	Alfanu- mérico	50			<i>Not null</i>	

Tabela FUNCIONARIO:

Atri- buto	Tipo	Tama- nho	Chave primá- ria	Chave estran- geira	Domí- nio	Default
CPF	Inteiro	11	X		<i>Not null</i>	
NMFUN- CIONA- RIO	Alfanu- mérico	100			<i>Not null</i>	
SEXO	Alfanu- mérico	1				F ou M
QTDE- FILHO	Inteiro	2				0
CDDE- PARTA- MENTO	Inteiro	5				

A criação do dicionário de dados seria realizada a partir da criação das tabelas:

```
Create table DEPARTAMENTO (  
    CDdepartamento int primary key not null,  
    NMdepartamento varchar (50) not null);  
Create table FUNCIONARIO (  
    Cpf int,  
    NMfuncionario varchar (100) not null,  
    Sexo char (1) check (sexo = 'F' or sexo = 'M'),  
    qtdeFilho smallint default 0,  
    CDdepartamento int,  
    primary key (cpf),  
    foreign key (CDdepartamento) references DEPARTAMENTO (CDdepartamento));
```

Depois da execução desses comandos de criação no banco de dados, o dicionário de dados englobaria os itens e aspectos listados a seguir.

- As tabelas DEPARTAMENTO e FUNCIONARIO.
- Para a tabela DEPARTAMENTO, os atributos (colunas) CDDEPARTAMENTO e NMDEPARTAMENTO, com suas características. O mesmo vale para a tabela FUNCIONARIO.
- Duas chaves primárias, CDDEPARTAMENTO para a tabela DEPARTAMENTO e CPF para FUNCIONARIO.
- Uma chave estrangeira na tabela FUNCIONARIO.
- As restrições de integridade *not null* para as colunas CDDEPARTAMENTO e NMDEPARTAMENTO, não permitindo que as colunas fiquem sem valor válido.
- As restrições de integridade *not null* para as colunas CPF e NMFUNCIONARIO. Para a coluna SEXO, só a possibilidade de conteúdo F ou M. Além disso, se o atributo QTDEFILHO não é informado, automaticamente é inserido o valor 0.

Todas essas regras são armazenadas nas tabelas do dicionário de dados para que, no momento de movimentação, seja checada a validade da operação.

Visualização

A qualquer momento, é possível visualizar as características dos dados, tanto a formação (como o tipo e o tamanho) quanto as regras que podem envolver determinado dado.

Suponha uma tabela CURSO que tenha um atributo denominado COD_CURSO representando a chave primária e uma tabela ALUNO que tenha um atributo denominado MATRICULA representando a chave primária. Por sua vez, COD_CURSO identifica o curso que o aluno frequenta, indicando uma chave estrangeira com a tabela CURSO. Para os atributos MATRICULA e COD_CURSO, ocorre a criação de restrição de integridade. Dessa forma, uma tabela **user_cons_columns** terá essas informações.

Tabela INFORMATION_SCHEMA TABLE_CONSTRAINTS:

Constraint_Catalog	Constraint_Schema	Table_schema	Table_name	Constraint_type	ENFORCED
def	Information_schema	Table_constraints	aluno	primary key	yes
def	Information_schema	Table_constraints	aluno	foreign key	yes
def	Information_schema	Table_constraints	curso	primary key	yes



Link

No *link* a seguir, você pode ver o detalhamento de cada atributo da tabela TABLE_CONSTRAINTS.

<https://qrqo.page.link/eKcm>

Como as informações estão armazenadas em tabelas, as consultas são realizadas dentro da mesma estrutura proposta para a visualização dos dados nas tabelas de negócio, por meio do comando `SELECT`. Você também pode relacionar as funções/rotinas definidas em um banco de dados:

```
SELECT ROUTINE_TYPE, ROUTINE_NAME  
FROM INFORMATION_SCHEMA.ROUTINES  
WHERE ROUTINE_SCHEMA='dbname';
```

O resultado dessa consulta é a apresentação do tipo e do nome da função que consta na tabela `INFORMATION_SCHEMA.ROUTINES`.



Saiba mais

Um dicionário de dados não é atualizado a partir dos comandos `INSERT`, `UPDATE` e `DELETE`. Sua formação acontece sempre a partir das definições/modificações das estruturas dos objetos. Caso aconteça a perda da área de algum objeto do dicionário de dados, a única forma de recuperação é por meio do procedimento `RESTORE`, baixa de *backup*, cópia de segurança.

Utilização

O uso do dicionário de dados no banco de dados é fundamental, pois determina a finalização da execução da transação. Entende-se por transação toda operação que se realiza no banco de dados (inclusão, alteração, exclusão e consulta). Dessa forma, a cada transação, é realizado um acesso às tabelas criadas no dicionário de dados para validação e disponibilidade da estrutura. O processo de validação confere a existência do objeto e suas características (domínio, restrição de integridade, índices).

Além disso, muitas tabelas criadas no dicionário de dados são usadas para a gestão do banco de dados. São utilizadas, por exemplo, para a verificação da quantidade de usuários e tabelas criadas; da incidência de índices em tabelas; e de indicadores que podem ser utilizados para melhor adequação de recursos no banco de dados, gerando melhor desempenho.



Referências

BITTENCOURT, R. G. *Aspectos básicos de banco de dados*. Florianópolis: [s. n.], 2004. Disponível em: <https://www.marilia.unesp.br/Home/Instituicao/Docentes/EdbertoFerneda/BD%20-%20Aspectos%20Basicos.pdf>. Acesso em: 17 abr. 2019.

HEUSER, C. A. *Projeto de banco de dados*. 6. ed. Porto Alegre: Bookman, 2009.

MYSQL data dictionary. In: ORACLE. *MySQL 8.0 reference manual*. c2019. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/data-dictionary.html>. Acesso em: 17 abr. 2019.

SILBERSCHATZ, A.; KORTH, H. F.; SUNDARSHAN, S. *Sistema de banco de dados*. Rio de Janeiro: Elsevier, 2012.

Leituras recomendadas

DATE, C. J. *Introdução a sistemas de bancos de dados*. Rio de Janeiro: Elsevier, 2004.

DUBOIS, P. *MySQL: language reference*. Indianapolis: MySQL Press, 2005.

RAMEZ, E.; NAVATHE, S. B. *Sistemas de banco de dados*. 6. ed. São Paulo: Person, 2011.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:

