

Objetivos:

- I. Diferença entre dados e informação;
- II. Vantagens da base de dados estruturada;
- III. Metadados;
- IV. BD - Banco de dados;
- V. SGBD - Sistema Gerenciador de Banco de Dados;
- VI. SQL - Structured Query Language - Linguagem Estruturada de Consulta;
- VII. Estrutura de um sistema de banco de dados;
- VIII. Características e funções do SGBD;
- IX. Tipos de SGBDs;
- X. Modelagem de banco de dados;
- XI. Passos de um projeto de BD.

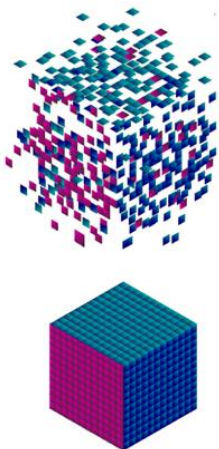
I. Diferença entre dados e informação

Dados X Informação



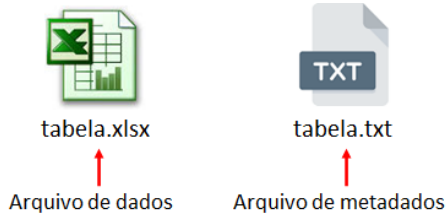
- Dados: são fatos num formato primário, que podem ser armazenados em um meio. Por exemplo: nome, endereço e idade;
- Informação: são os fatos organizados de maneira a produzir um significado, ou seja, são dados colocados em um contexto. Por exemplo, os dados dos alunos podem ser categorizados por faixa etária e curso para sabermos se existe alguma tendência de idade por curso. Fora de um contexto esses dados não fazem sentido.

II. Vantagens da base de dados estruturada



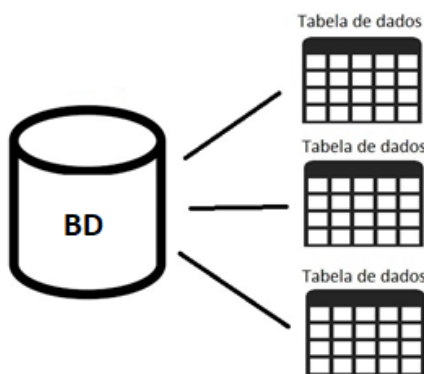
- Os dados devem ser organizados e estruturados para que possam ser usados com eficácia;
- A má organização de dados dificulta a sua localização e acesso;
- As vantagens de uma boa organização de dados ficam evidentes à medida que a quantidade e variedade de dados aumenta;
- Ainda é comum termos dados armazenados em arquivos textos e planilhas, mas esses formatos não ajudam na localização e cruzamento de dados.

III. Metadados



- Metadados são “dados sobre os dados”;
- Eles são usados para descrever o significado de um conjunto de dados. Por exemplo, numa tabela tem-se somente as colunas e linhas. Para dar mais clareza para o usuário podemos colocar num arquivo separado as descrições das colunas. Esse arquivo é um metadado da tabela e será usado apenas no caso de dúvidas/esclarecimentos.

IV. BD - Banco de Dados

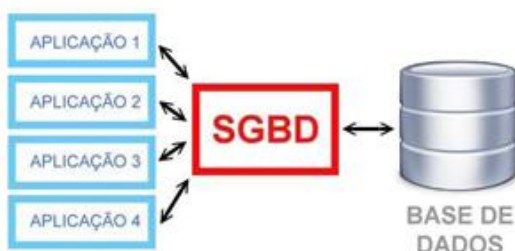


- Um BD é formado basicamente por tabelas;
- Um BD é como se fosse uma pasta no computador e as tabelas são como arquivos nessa pasta;
- Uma tabela é formada por colunas;
- Os dados estão nas linhas da tabela;
- Cada coluna possui um nome e tipo de dado que a coluna aceita. Por exemplo:
 - A coluna nome aceita caracteres alfanuméricos;
 - A coluna idade só aceita valores inteiros;
 - A coluna peso só aceita valores do conjunto dos Reais.

Tabela

nome (alfanumérico)	idade (inteiro)	peso (real)
Ana	22	59.5
Pedro	25	70.2

V. SGBD - Sistema Gerenciador de Banco de Dados

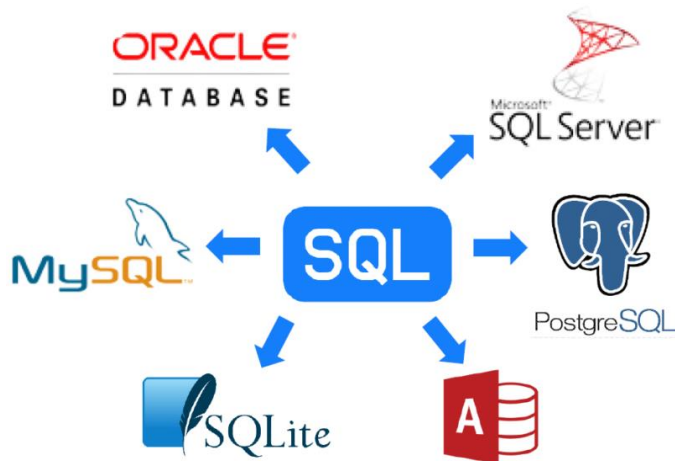


- Um SGBD é um conjunto de programas para armazenar, gerenciar e consultar bases de dados;
- Um BD é como se fosse uma pasta no computador e o SGBD é o software responsável por criar e fazer todas as operações nessa pasta;
- Exemplos de SGBD: Oracle, SQL Server, MySQL, PostgreSQL, MongoDB, SQLite etc.
(<https://becode.com.br/principais-sgbds>).

Estritamente falado, o termo BD deve ser aplicado apenas aos dados, enquanto o termo SGBD deve ser aplicado ao software com a capacidade de manipular BD. Porém, é comum misturar os dois conceitos.

O Stack Overflow é o principal site de dúvidas utilizado pelos programadores. Anualmente eles fazem um ranking dos SGBDs mais pesquisados. Veja o ranking de 2021 em <https://insights.stackoverflow.com/survey/2021#section-most-popular-technologies-databases>.

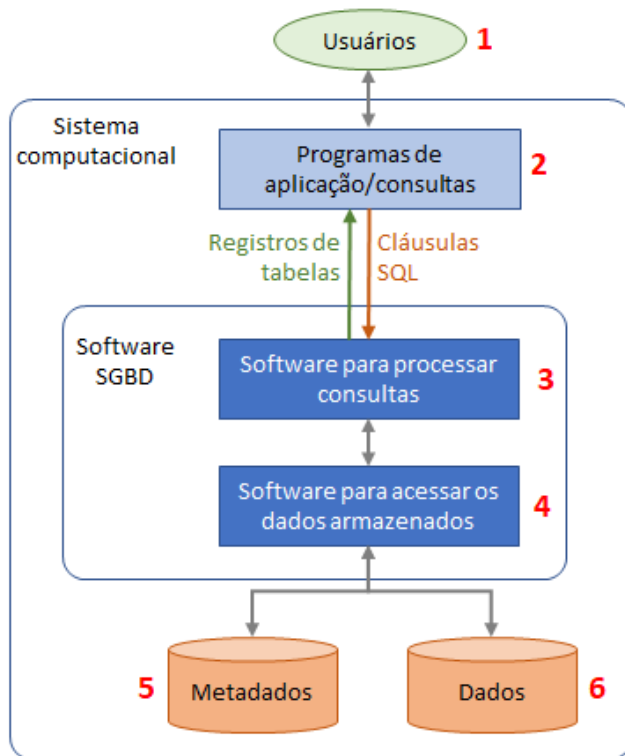
VI. SQL - Structured Query Language - Linguagem Estruturada de Consulta



- O SGBD é um software que não possui uma interface gráfica, ou seja, não podemos acessar ele usando botões e outras facilidades das interfaces de usuário;
- O SGBD responde somente a comandos escritos na linguagem SQL, logo precisaremos aprender SQL para nos comunicar com o SGBD;
- Os comandos SQL possuem uma estrutura mais simples que as linguagens de programação.

VII. Estrutura de um sistema de banco de dados

Um sistema de BD é um sistema computacional que envolve usuários e dados armazenados em SGBD.



1. Usuários: são os programadores e usuários. Considere como exemplo, o SIGA – nosso sistema acadêmico – os alunos e professores são usuários do SIGA e não interagem diretamente com o SGBD;
2. Programas de aplicação/consultas: o SIGA é um programa, codificado em alguma linguagem de programação, que faz acesso ao SGBD;
3. Software para processar consultas: o SGBD interpreta os comandos SQL para saber aquilo que precisa ser feito;
4. Software para acessar os dados armazenados: o SGBD resgata os dados armazenados no sistema de armazenamento;
5. Metadados: são informações utilizadas por (4) para ajudar na identificação e resgate dos dados;
6. Dados: os dados propriamente dito. Observe que os dados e metadados ficam fora do SGBD.

O SGBD isola os usuários de detalhes a nível de hardware - como os dados estão armazenados.

Toda a comunicação com o SGBD se dá através de comandos (cláusulas) SQL e a resposta do SGBD são registros de tabelas, pois todos os dados estão armazenados em tabelas.

VIII. Características e funções do SGBD

- Controle de redundância: as estruturas de armazenamento são modeladas para evitar duplicidade de dados, isto é, evitar o armazenamento do mesmo dado várias vezes;
- Precisão: dados precisos, consistentes são um sinal de integridade dos dados. SGBDs fomentam a integridade dos dados, porque as atualizações e alterações dos dados precisam ser realizadas em um só local. As chances de se cometer um erro são maiores se existir a necessidade de alterar os mesmos dados em vários lugares diferentes do que se você só tem que fazer a mudança em um só lugar;

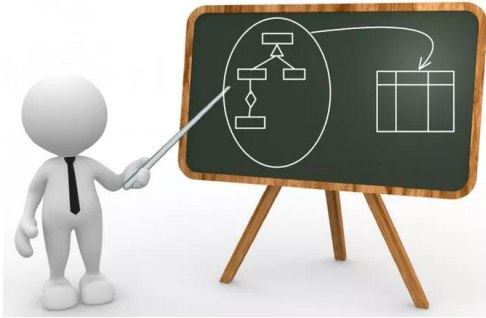
- Controle de concorrência: em sistemas com muitos usuários pode ocorrer de um mesmo registro ser acessado por vários usuários ao mesmo tempo, por exemplo, pode gerar inconsistência se várias operações alterarem a quantidade de itens de um produto no estoque;
- Consistência no sistema de arquivos: o SGBD padroniza os formatos de tabelas. Isso faz com que os tabelas de dados sejam mais fáceis de manter, porque as mesmas regras e diretrizes se aplicam a todos os tipos de dados. O nível de consistência entre as tabelas e programas (softwares que consomem os dados) torna mais fácil de gerenciar dados quando vários programadores estão envolvidos;
- Independência de dados: o SGBD dá aos usuários uma visão abstrata dos dados, encobrindo detalhes não relevantes. O programador não precisa saber como os dados estão fisicamente armazenados;
- Backup e restauração: um SGBD pode ser configurado para manter cópias de segurança, inclusive persistindo os dados num SGBD remoto;
- Múltiplas visões dos dados: o mesmo dado pode ser exibido de diferentes formas, isto é, cada perfil de usuário pode ver uma parte distinta dos dados;
- Autenticação e autorização de acesso: o SGBD só permite o acesso de usuários autorizados, inclusive é possível limitar o acesso a somente partes dos dados para cada usuário;
- Restrições de integridade: como exemplo, o SGBD garante que uma coluna que aceita números só poderá receber números, ou seja, é mantida a integridade dos dados nessa coluna.

IX. Tipos de SGBDs

- Os primeiros SGBDs possuíam modelos conhecidos como hierárquicos e em rede. Estes SGBDs caíram em desuso devido às suas limitações e problemas em seus modelos;
- Atualmente, o mercado para SGBDs concentra-se nas tecnologias de SGBDs Relacionais (SGBD-R) e SGBDs Orientados-a-Objeto (SGBD-OO). Sendo que os SGBDs Objeto-Relacionais incorporaram a capacidade de criar tipos objetos (SGBD-OR). Exemplos de SGBDs:
 - SGBD-OR: Oracle, SQL Server, MySQL, PostgreSQL e SQLite;
 - SGBD-OO: MongoDB.

X. Modelagem de banco de dados

O produto da modelagem de banco de dados é o modelo de banco de dados.



É o processo de **levantar**, **analisar**, **organizar** e **armazenar** todos os dados da aplicação:

- **Levantar** significa identificar todos os dados necessários e disponíveis, não faz sentido criar um modelo de dados sem conhecer as características dos dados: tipo de dado (integer, float, varchar, date etc.), volume (quantidade estimada de registros) e uso (entender como o dado será utilizado);
- **Analisar** significa identificar os dados que serão relevantes para a aplicação. O cliente pode apresentar diversos dados e alguns deles não serem relevantes para a aplicação ao mesmo tempo que deixa de apresentar dados importantes. Manter dados irrelevantes na base de dados torna a gestão dos dados complexa desnecessariamente;
- **Organizar** significa distribuir os dados nas tabelas de modo a evitar redundâncias e perdas de dados. Perder dados significa deixar de manter na base de dados um dado que será útil para a aplicação no futuro;
- **Armazenar** significa criar as tabelas no ambiente do SGBD.

Os modelos de banco são usados para descrever, mais detalhadamente, a estrutura de um banco de dados. Os modelos podem ser apresentados nos níveis **conceitual**, **lógico** e **físico**.

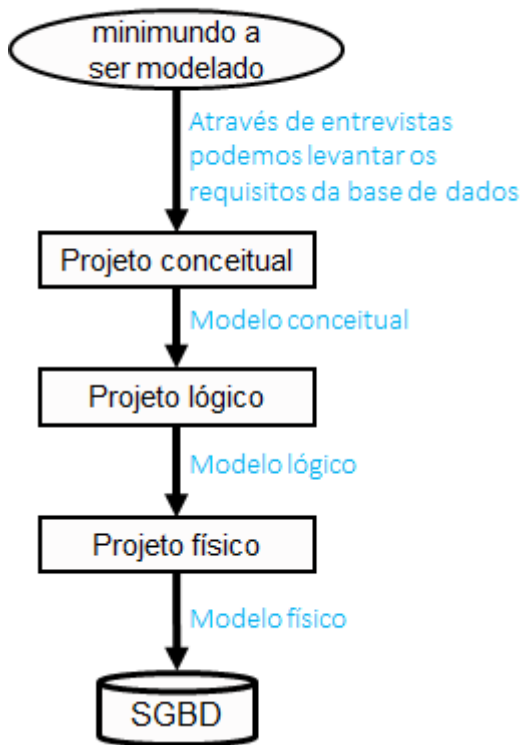
XI. Passos de um projeto de BD

A modelagem de dados consiste na análise e planejamento dos dados que irão compor o banco. Ela tem início nas entrevistas com o cliente para obtermos as informações a serem armazenadas no BD.

Num projeto de BD podem ser feitos os modelos **conceitual**, **lógico** e **físico**. Os modelos conceitual e lógico são semelhantes, por isso é comum partir para o lógico sem passar pelo conceitual. Ambos são gerais e são pensados sem preocupação com eficiência ou no SGBD que será implementado. O conceitual é um pouco mais geral que o lógico. O modelo físico é mais detalhado e considera o SGBD que o modelo vai rodar.

Para efeito de desenvolvimento da aplicação o físico não é muito considerado em primeiro momento. Em alguns casos o físico pode ser igual ao lógico, em termos de estrutura, mas pode precisar mudar para atender a necessidade do SGDB ou para alcançar alguma meta de eficiência. O físico é o que importa no momento de implementar.

Passos de um projeto de BD



Levantamento e análise de requisitos

Ela tem início nas entrevistas com o cliente para obtermos as informações a serem armazenadas no BD.

É necessário compreender:

- ✓ Quais dados serão armazenados;
- ✓ Quais aplicações utilizarão esses dados;
- ✓ Quais são as operações mais frequentes com os dados.

O objetivo é entender o que os usuários desejam do BD. O resultado é um conjunto de requisitos que serão usados na construção do modelo conceitual do BD.

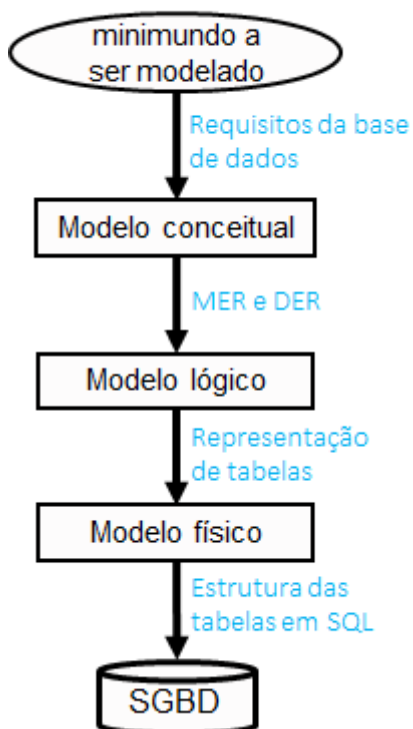
Projeto conceitual do BD

As informações obtidas na análise de requisitos são usadas para desenvolver uma descrição de alto nível dos dados.

Essa descrição é expressa no MER (Modelo Entidade Relacionamento) e DER (Diagrama Entidade Relacionamento).

O objetivo é facilitar o entendimento e discussão entre os envolvidos. O resultado é o DER.

Passos de um projeto de BD



Projeto lógico do BD

Converter o DER para um esquema de tabelas, incluindo informações de tipos de dados e chaves. A maior diferença entre os modelos conceitual e lógico, é que o lógico tem dados mais técnicos, por exemplo o tipo do dado, o tamanho que ele comporta, e eventualmente alguma restrição de como o dado pode ser. O conceitual só precisa saber quais são os dados de forma geral sem pensar em nada técnico, precisamos apenas saber quais os atributos da entidade e como as entidades se relacionam.

O objetivo é converter de ER para o modelo relacional. O resultado é um modelo de tabelas.

Projeto físico do BD

Converter o modelo relacional para cláusulas SQL de acordo com o SGBD escolhido para implantar o BD. No modelo físico estará tudo o que é necessário para criar as tabelas no SGBD,

não só as colunas, mas também chaves, índices, gatilhos e restrições técnicas - inclusive de permissões de acesso.

O objetivo é converter de modelo relacional para cláusulas SQL. O resultado é o código SQL para criar as tabelas.

A figura a seguir mostra a entidade Aluno, apesar dessa entidade ter inúmeros atributos no mundo real, no mundo acadêmico bastaria o RA e nome (modelo conceitual). O segundo passo é representar essa entidade num modelo de representação, aqui usamos o modelo relacional (modelo lógico) e o terceiro passo é criar essa representação na linguagem SQL compatível com o SGBD escolhido (modelo físico).

Minimundo a ser modelado



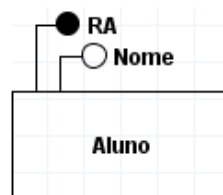
RA: 123

Nome: Pedro

Modelo conceitual

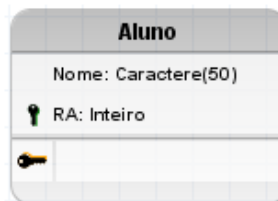
Aluno: RA e Nome

MER



DER

Modelo lógico



Representação de uma tabela

Modelo físico

```
create table tbaluno(
    ra integer not null primary key,
    nome varchar(50)
);
```

Cláusula SQL compatível com o SGBD PostgreSQL