BANCO DE DADOS

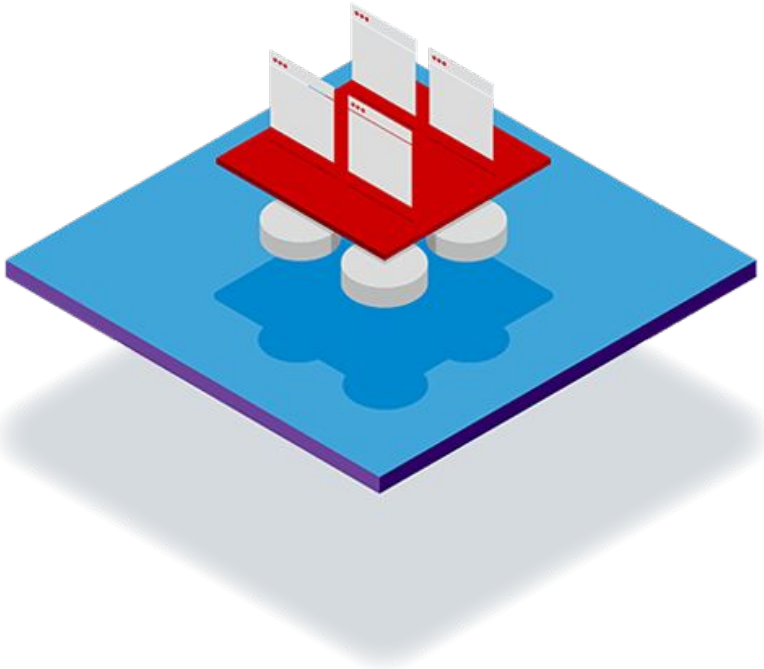
programa}

WHO AM I?

DESSA FREIRES

- ★ Apaixonada por pessoas, diversidade e dados!
- ★ Palestrante - Data Science / Diversidade
- ★ Cientista de Dados, Representante de Inclusão, Diversidade e Cultura, Mentora - PicPay
- ★ Coordenadora: UneAFRO e AI Girls
- ★ Colunista - Female Tech Leaders
- ★ Produtora de conteúdo: @datasciencedescomplicada @diversidadedescomplicada
- ★ Voluntária/Membro: Afropython, Pyladies, AIGirls, UXPMP, BOSS
- ★ Instagram: @dessafreires
- ★ Linkedin: https://www.linkedin.com/in/andressafreires/
- ★ Whatsapp: 11943826870

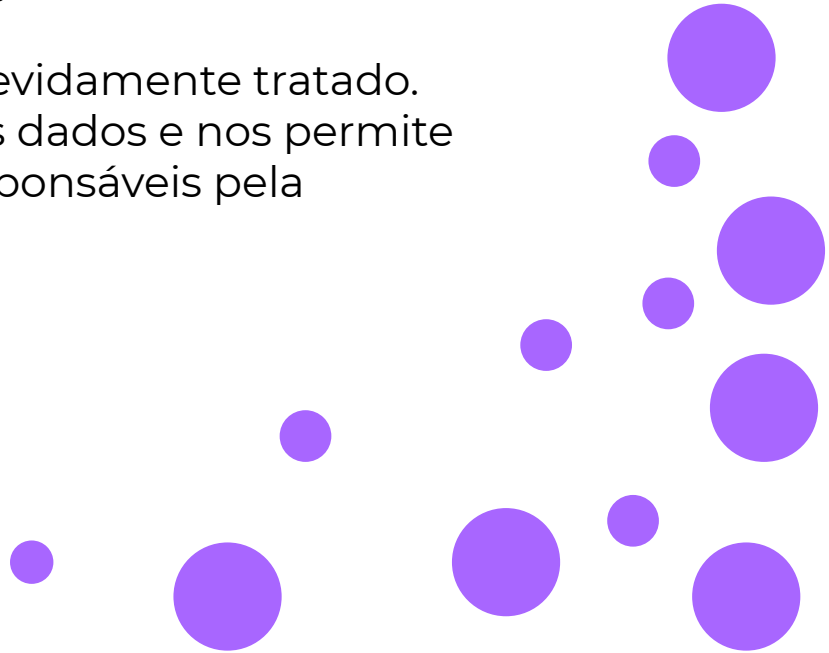
Data-driven: Por que as empresas estão buscando ser assim?



Dados: O que são?

Tudo o que fazemos hoje em dia gera **dados**, que são elementos que constituem a matéria-prima das informações.

Dado é um conhecimento bruto, ainda não devidamente tratado. Já a informação nos dá a interpretação desses dados e nos permite **obter insights**. Os insights são os maiores responsáveis pela inovação.

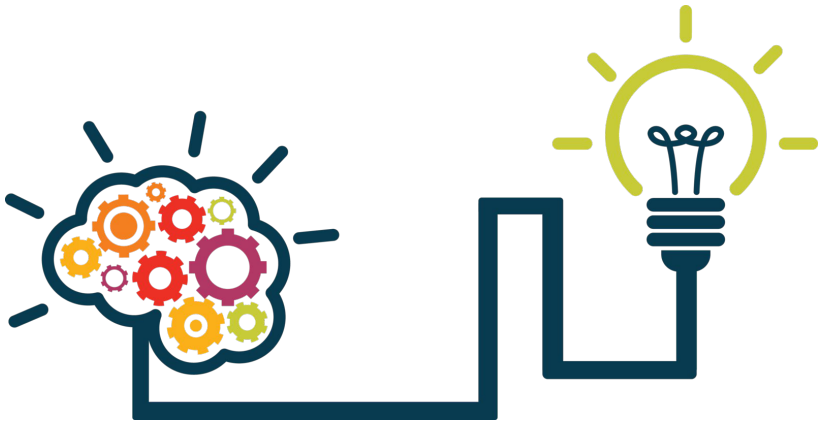


O que é cultura Data-driven?

A cultura data driven prioriza a **tomada de decisão baseada em dados!**

Quanto mais informações estiverem acessíveis aos decisores e quanto mais detalhadas forem essas informações, **mais eficiente e assertivo** será o processo de decisão.

A importância de ter contexto do problema.



O contexto: como tê-lo?

Para que a gente consiga chegar em boas soluções provindas de dados, não devemos olhar direto para os dados em si. Devemos primeiro entender o **contexto** do problema.

Sem o devido contexto, os dados não servem para **NADA**. Procure sempre responder essas questões para entender como prosseguir:

1 – Quais informações são relevantes para o andamento da solução?

2 – Quem vai usufruir da sua solução? Para que(m) ela será útil?
O que você sabe a respeito dos usuários?

3 – Quais dados reforçam o que você quer passar? O público tomador de decisão e usuário os conhece?

4 – Onde estão os riscos? Que fatores podem vir a ser impeditivos?

5 – Quais recursos você tem para alcançar seu objetivo? Eles são suficientes? Como você irá medir isso?

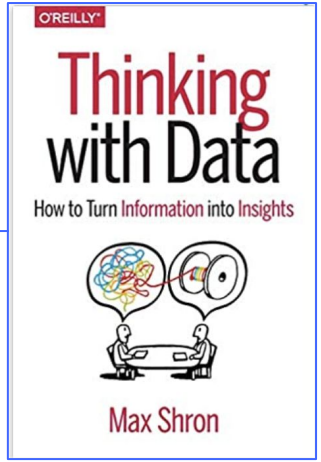
Pensamento focado em dados.

Por que pensar com foco em dados?

PARA ENTENDER, É PRECISO PERGUNTAR!

Perceba que as respostas de cada uma das perguntas anteriores nos levam a pensar em **dados**. E você precisa saber usá-los a seu favor!

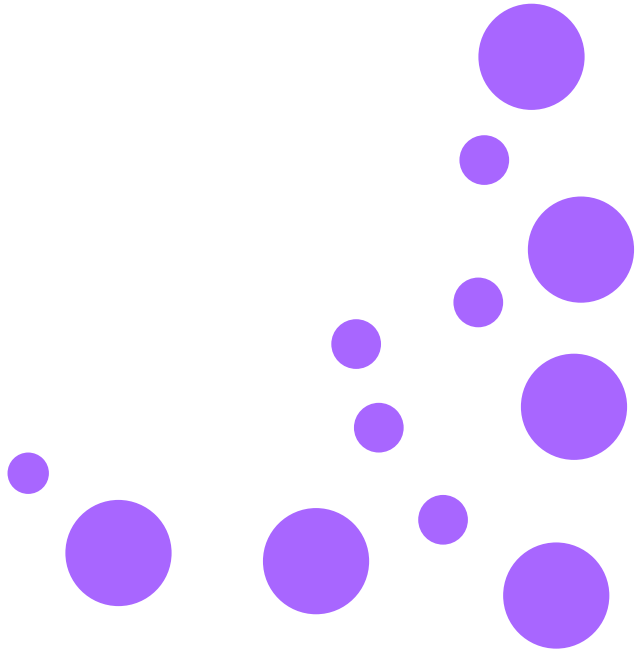
O que você precisa fazer? **ORGANIZAR** seus dados, transformá-los em informação e transformar as informações em ideias factíveis.



Por que pensar com foco em dados?

Os benefícios de orientar a tomada de decisão com dados são inúmeros:

- Redução de custos;
- Diminuição do retrabalho;
- Eficiência;
- Satisfação do cliente;
- Valor de mercado.



Uso de dados como ferramenta para negócios.

Dados + negócio = 💜

Em geral, quanto mais você entender as nuances do negócio, mais direcionadas serão suas previsões e, no final das contas, melhores insights você encontrará.

Portanto, faça tudo o que puder para entrar nos **detalhes do negócio**. Procure colegas que entendem de negócio, aprenda com eles e, se possível, torne-os seus conspiradores.

*OBS: Obviamente, é bom ter conhecimento de dados, mas seu conhecimento do negócio terá um **impacto muito maior** na qualidade do seu trabalho.*

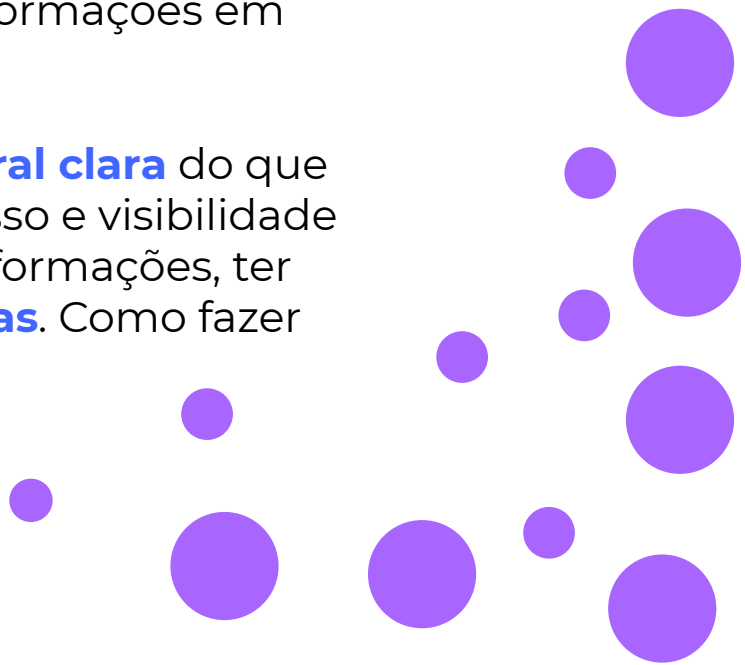
Como tirar insights a partir de dados?

Dados = insights?

Infelizmente, **NÃO**!

Como dito antes, é necessário **ORGANIZAR** seus dados, transformá-los em informação e transformar as informações em insights.

Dados organizados podem fornecer uma **visão geral clara** do que está acontecendo em sua empresa. Com fácil acesso e visibilidade aos dados organizados, é mais fácil processar as informações, ter ideias e tomar **decisões mais inteligentes e rápidas**. Como fazer isso?



Transformando dados em insights

- Concentre-se em **tendências**, não em pontos de dados: os melhores insights geralmente vêm não de olhar para pontos de dados singulares, mas de tendências.

- **Compare** os intervalos de tempo: examine os diferentes intervalos de tempo, como semana após semana, mês após mês ou verão após verão.

- **Procure relacionamentos** fortes: Frequentemente, as descobertas mais poderosas e perspicazes na análise de dados são os relacionamentos entre dados.

- Experimente **diferentes perspectivas**: como um indivíduo não consegue ver tudo, convide outras pessoas para se aprofundarem nos dados. É vital ter vários pares de olhos procurando percepções acionáveis.

- **Seja cético**: sempre analise os dados de diferentes ângulos. Por exemplo, plote os mesmos dados várias vezes usando diferentes tipos de gráfico. Os dados têm o poder de enganar, por isso certifique-se de que contam a história com precisão.

BANCO DE DADOS

Banco de Dados - Conceitos

- **Dado**: é o componente básico de um arquivo, é um elemento com um significado no mundo real, que compõe um sistema de arquivos. Como exemplo, podemos citar nome, sobrenome, cidade, bairro e outros.

- **Informação**: após a interpretação dos dados, é possível associar um significado aos dados ou processá-los. Normalmente a informação vem de convenções utilizadas por pessoas por meio de associações aos dados.

- **Conhecimento**: todo discernimento, obtido por meio de critérios, e apreciação aos dados e informações.

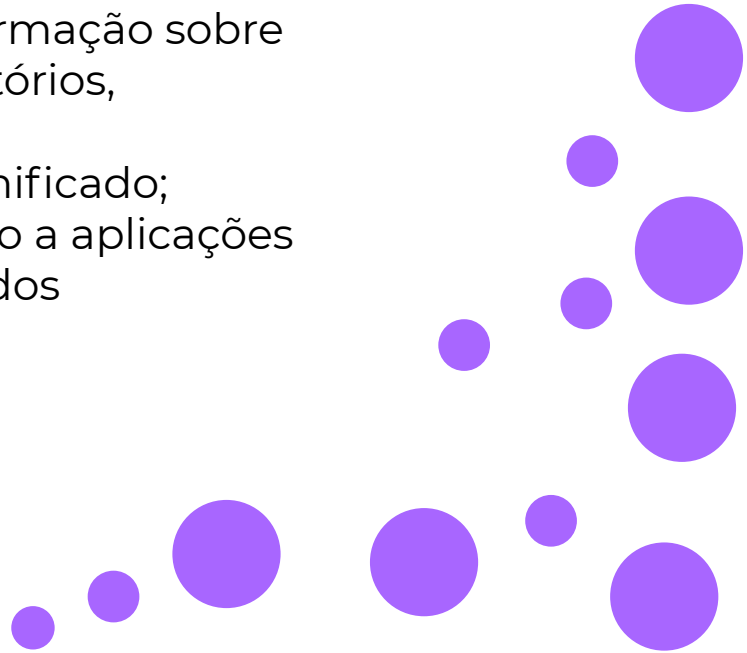
Banco de Dados - Conceitos

- “Um Banco de Dados é um conjunto de arquivos relacionados entre si” (Chu, 1983)
- “Um Banco de Dados é uma coleção de dados operacionais armazenados, sendo usados pelos sistemas de aplicação de uma determinada organização” (C. J. Date, 1985)
- “Um Banco de Dados é uma coleção de dados relacionais” (Elmasri e Navathe, 2005).
- “Um Banco de Dados é um objeto mais complexo, é uma coleção de dados armazenados e inter-relacionados, que atende às necessidades de vários usuários dentro de uma ou mais organizações, ou seja, coleções inter-relacionadas de muitos tipos diferentes de tabelas.” (TOREY et al,p.2,2007)”

Banco de Dados - Conceitos

Com base nas definições da literatura, podemos dizer que Banco de Dados é:

- Coleção de dados relacionados que têm informação sobre algo do mundo real, por exemplo, lojas, escritórios, bibliotecas ou banco;
- Possui coerência lógica entre de dados e significado;
- Um Banco de Dados sempre estará associado a aplicações onde existem usuários com interesse aos dados relacionados.



Banco de Dados - Conceitos

É importante entender que você pode ter um Banco de Dados utilizando as tecnologias computacionais ou não.

Veja bem, uma lista telefônica é bom exemplo de um Banco de Dados que não utiliza informatização, porém ela possui todos os dados de pessoas, empresas com números de telefone onde podemos consultar por meio de sobrenomes ou índices (ELMASRI e NAVATHE, 2011).

Quais são as maiores dificuldades encontradas para encontrar os dados armazenados em tais bancos?

Sistema De Gerenciamento De Banco de Dados

Um Sistema Gerenciador de Banco de Dados (SGBD) é um software genérico para manipular Banco de Dados.

Ele permite a definição, construção e manejo de um Banco de Dados para diversas aplicações.

Um SGBD possui uma visão lógica (projeto do BD), uma linguagem de definição de dados, linguagem de manipulação de dados e utilitarios importantes.

**Uma linguagem de definição de dados (DDL) é usada para definir o esquema conceitual do Banco de Dados.*

Sistema De Gerenciamento De Banco de Dados

Com a utilização de um Banco de Dados temos seguintes vantagens:

- **Controle centralizado de dados**: os dados ficam localizados em um único local e isso facilita o controle e acesso;
- **Controle da redundância**: como os dados estão centralizados existe otimização e redução do espaço de armazenamento e controle de redundância;
- **Compartilhamento de dados**: com a utilização de um Banco de Dados sem redundância, o espaço de armazenamento é otimizado é facilita o compartilhamento de dados;
- **Facilidade de acesso aos dados**: estabelecimento de padrões devido a centralização dos dados e inter-relação de todos os registros;

Sistema De Gerenciamento De Banco de Dados

Exemplos:

- **Microsoft SQL Server**: O Microsoft SQL Server é uma plataforma abrangente que fornece sofisticadas ferramentas de gerenciamento de dados e integração de Business Intelligence.

  O SQL Server oferece alta performance em aplicações críticas, utilizando tecnologias in-memory, fluxos rápidos de informações em ferramentas familiares como o Excel.

  Além disso, a Microsoft oferece bons descontos para instituições educacionais, incluindo escolas e universidades públicas, o que contribuiu para a constituição de uma base sólida de usuários. Se a sua empresa necessita de um bom suporte para a recuperação de dados, o SQL Server é uma excelente escolha.

Sistema De Gerenciamento De Banco de Dados

Exemplos:

- **Oracle RDBMS**: A última versão do Oracle, chamada de 18c (o “c” faz referência à cloud computing) e projetada para computação em grade, é o melhor sistema gerenciador de bancos de dados relacional (RDBMS, na sigla inglesa).

  A Oracle possui um inabalável posicionamento no mercado, oferecendo soluções para grandes corporações, agências governamentais e data centers que operam em arquitetura RAC.

  É a opção ideal para negócios que necessitam de alta disponibilidade, segurança contra falhas nos servidores e escalabilidade (sendo possível adicionar mais nodes conforme a demanda).

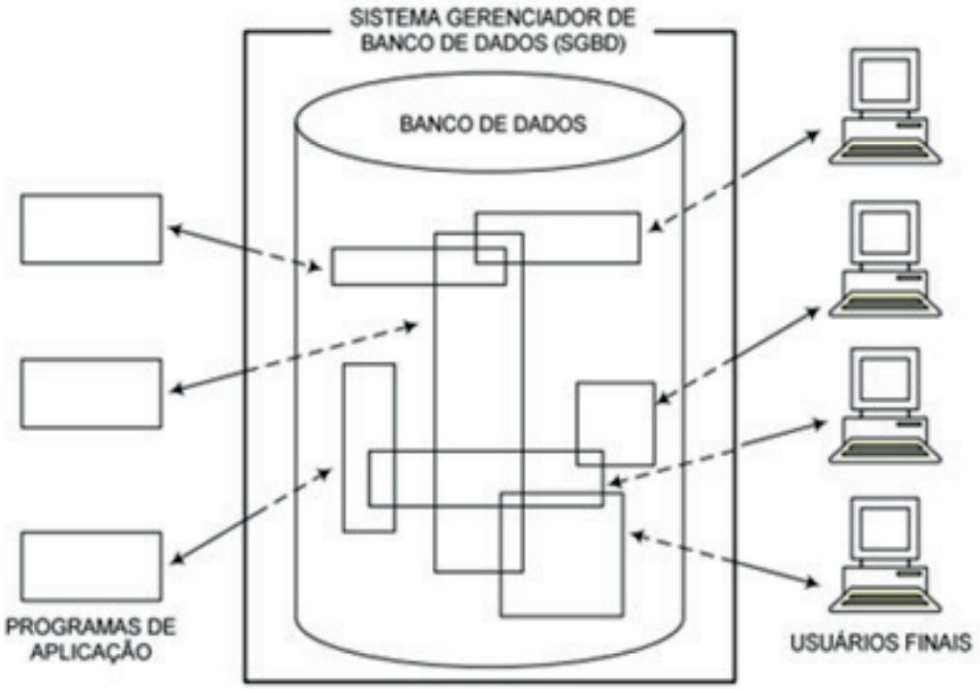
Sistema De Gerenciamento De Banco de Dados

Os SGBDs também facilitam a conversão e a reorganização do Banco de Dados.

Dessa forma podemos dizer que os SGBDs são programas de computador, que ajudam na:

- Definição e construção do Banco de Dados: que é o processo de criar uma estrutura inicial com tabelas e preenchê-las com dados;
- Manipulação dos dados do Banco de Dados: são as operações de consultas alteração, exclusão realizadas nos dados.

Sistema De Gerenciamento De Banco de Dados



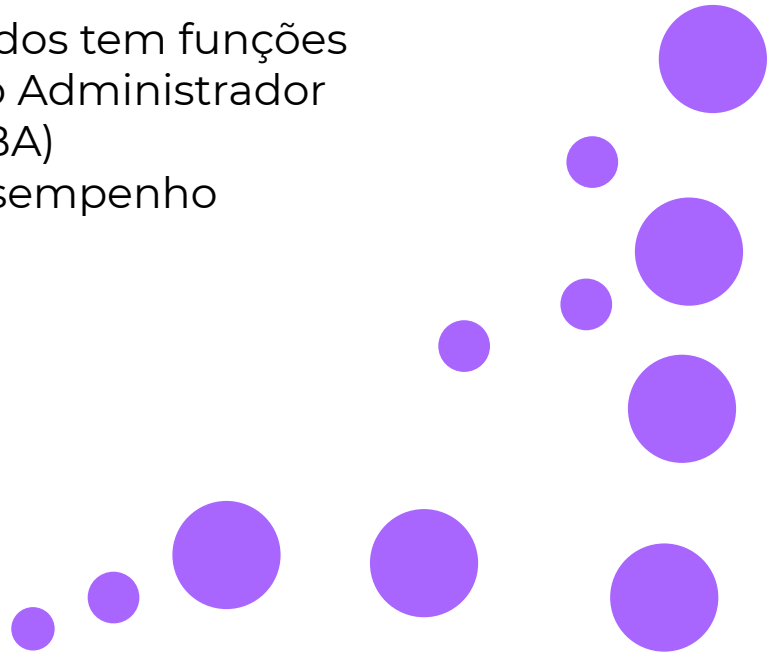
Aplicações de Banco de Dados

- Banco: para informações de cliente, contas, empréstimos e transações bancárias;
- Universidades: informações de alunos, cursos e notas;
- Linhas Aéreas: reservas e informações de horários;
- Indústria: controlar a produção de itens da fábrica, estoques e pedidos.
- Vendas: informações de cliente, produto e compras;
- Biblioteca: informações de livros, nome dos alunos e solicitações

Visões Do Banco de Dados

Aplicações de Banco de Dados para se tornarem seguras e eficientes tornam-se complexas.

Dessa forma os Administradores de Banco de Dados tem funções que devem facilitar a vida do usuário. É função do Administrador de Banco de Dados - DataBase Administrator (DBA) garantir que os dados estejam seguros e com desempenho satisfatório.



Visões Do Banco de Dados

Esse profissional é responsável por garantir (HEUSER, 2004):

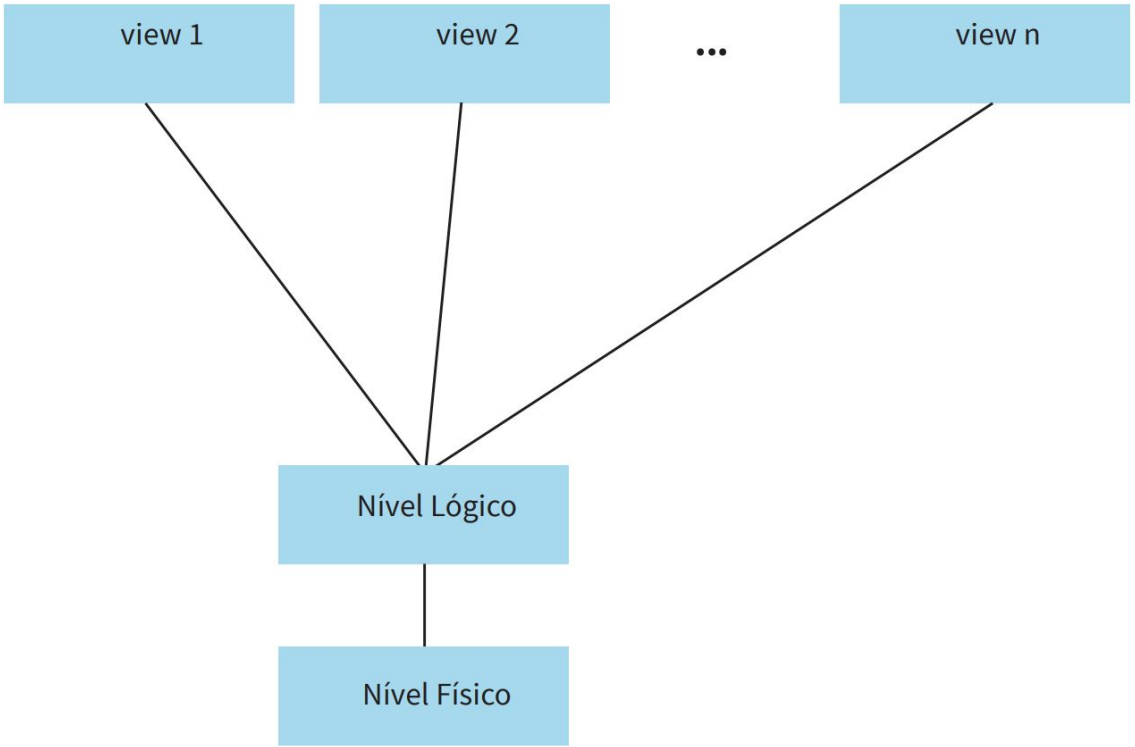
- **Segurança do Banco de Dados**: garantir segurança aos dados, permitindo que apenas usuários com acesso adequado possam utilizar o banco.
- **Recuperação**: deve sempre realizar procedimentos de backup para evitar que falhas façam que ocorra perda dos dados.
- **Disponibilidade**: ele tem responsabilidade de manter o Banco de Dados sempre disponível.
- **Suporte a equipe de desenvolvimento**: Bom relacionamento entre a equipe e o DBA para desenvolvimento e manutenção.
- **Implementação de Bancos de Dados**: Deve sempre realizar de forma adequa a implementação do banco.

Visões Do Banco de Dados

Uma das funções essenciais ao DBA é utilizar abstração de dados. Com a utilização de abstração de dados, é possível esconder certos “detalhes” sobre como os dados estão armazenados e como é realizada a manutenção, para facilitar o entendimento do usuário (JUKIC, VRBSKY S e NESTOROV, 2013). Os Níveis de Abstração são:

- **Nível físico**: Descreve como os dados estão armazenados. Este é o nível mais baixo de abstração.
- **Nível lógico**: Esse nível de abstração está acima do físico e descreve quais dados estão armazenados no BD e quais são suas relações. Descreve o Banco de Dados inteiro em termos de um pequeno número de estruturas relativamente simples.
- **Nível visões**: Esse nível pode ser visto pelo usuário de diversas formas pois quem opera são os sistemas aplicativos. Esse nível existe para facilitar sua interação com o sistema, ou seja, o sistema pode fornecer muitas visões para o mesmo Banco de Dados.

Visões Do Banco de Dados



Projeto de Banco de Dados

Os sistemas de Banco de Dados tem um ciclo de vida para sua execução. ELMASRI e NAVATHE (2011) elencam as etapas em oito fases:

1. **Definição do sistema**: nesta etapa é determinado o escopo do sistema, o deverá ser armazenado, e quais serão as operações realização assim como seus usuários.

2.**Projeto do Banco de Dados**: esta etapa é a criação do projeto conceitual, lógico e físico.

3.**Implementação do Banco de Dados**: cria-se realmente o Banco de Dados, conforme esquemas definidos na etapa anterior.

4.**Carga ou conversão de dados**: nesta etapa, o Banco de Dados é preenchido com dados já existes ou é preenchido manualmente.

Projeto de Banco de Dados

5.**Conversão de aplicação**: nesta etapa os programas que antes acessavam o banco, são informados sobre as modificações do o novo banco (etapa auxiliar).

6.**Teste e validação**: verifica-se tudo está funcionando de acordo com o planejamento.

7.**Operação**: é relativo à disponibilização do sistema para uso;

8.**Monitoramento e manutenção**: nesta etapa observa-se o funcionamento do sistema para possíveis ajustes.

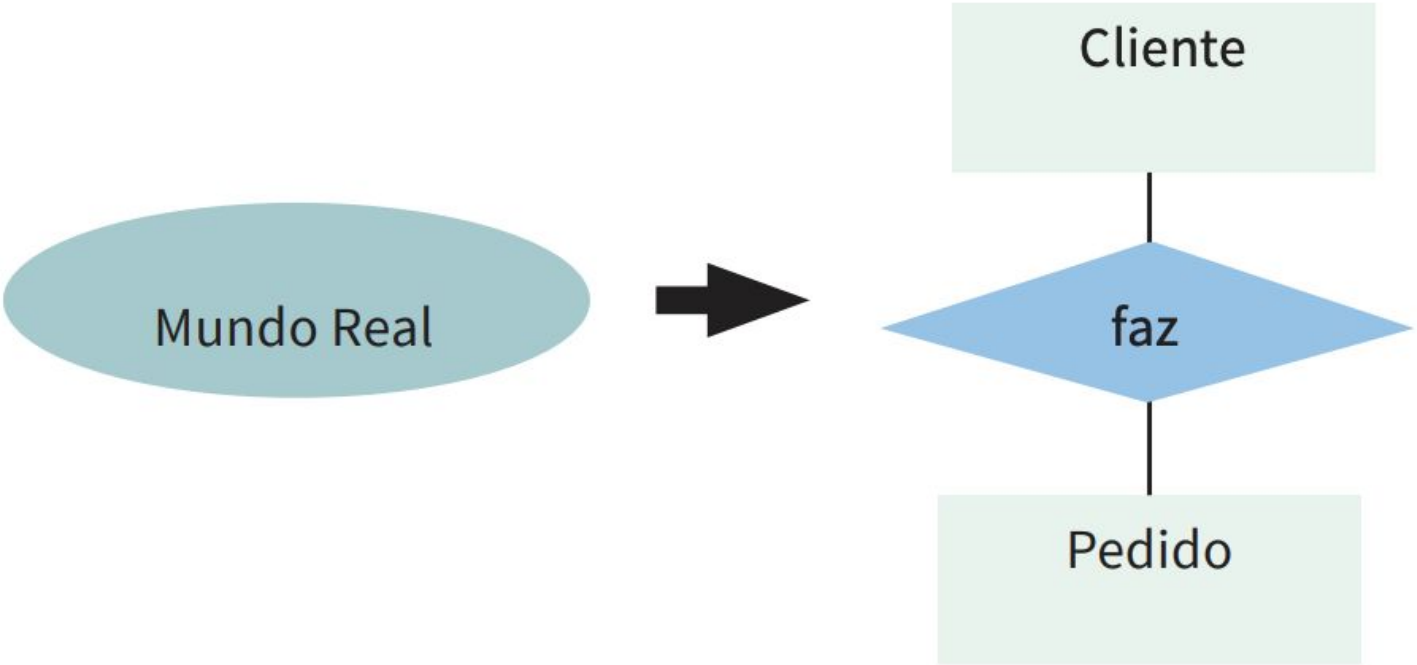
Modelo Conceitual

Em um projeto de Banco de Dados normalmente utiliza-se o Modelo Entidade Relacionamento para descrever quais são os requisitos que o usuário deseja no Banco de Dados (GARCIA-MOLINA, H.; ULLMAN, J. D.; WIDOM, 2008).

No projeto conceitual é desenvolvido um modelo de alto nível para atender os requisitos elencados pelo usuário (solicitante do banco) durante a definição do sistema (CORONEL, MORRIS, e ROB, 2012).

Dessa forma o modelo conceitual irá descrever a realidade do ambiente real e o problema, sendo uma visão geral dos principais dados e suas relações, independente das restrições de implementação (JUKIC, N.; VRBSKY S.; NESTOROV, S, 2013).

Modelo Conceitual



É possível observar o modelo Entidade Relacionamento de uma empresa onde o Cliente faz um pedido, é realizada uma **abordagem conceitual de como de quais serão as entidades retratadas no Banco de Dados e quais serão suas relações. O Modelo Conceitual não irá dizer como será implementado o Banco de Dados por um SGBD tanto em forma física quanto em forma lógica.*

Modelo Lógico

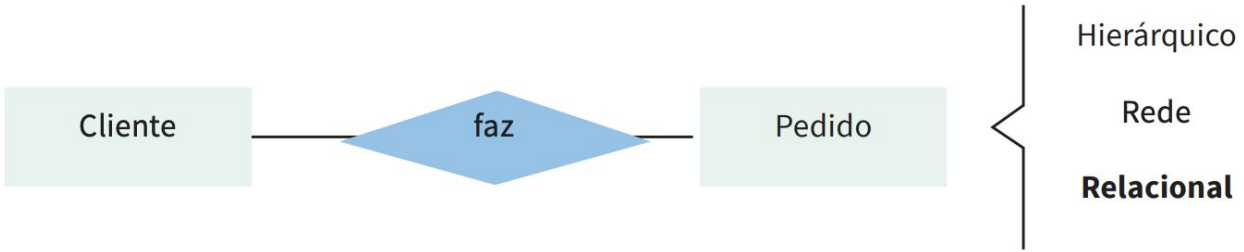
O Modelo Lógico tem três abordagens atualmente possíveis:

- **Modelo relacional**: classifica os dados em tabelas que possui colunas e linhas, que possuem relacionamentos. Entre os três modelos apresentados, o mais utilizado é o modelo Relacional.
- **Modelo hierárquico**: a organização é feita como uma árvore, onde cada registro tem um único “pai” e registros “irmãos” são colocados em uma ordem específica(CORONEL, MORRIS, e ROB, 2012).
- **Modelo de rede**: Cada conjunto consiste em um registro proprietário, e um ou mais registros de membro. Um registro pode ser um membro, em vários conjuntos, permitindo que esse modelo transmita relações complexas (HEUSER, 2004).

Modelo Lógico

A partir do modelo conceitual, ou seja, após a elaboração do modelo Entidade Relacionamento é possível ter uma visão geral de como será o sistema de Banco de Dados.

O modelo lógico irá descrever quais serão as estruturas que devem conter no Banco de Dados, sem considerar características específicas do SGBD, ou seja, não há preocupação em qual software será utilizado até este momento.



Modelo Físico

O Modelo Físico irá descrever as estruturas físicas de armazenamento de dados, como (HEUSER, 2004):

- Tamanho de campos;
- Tipos dos campos;
- Terminologia dos campos, que serão projetadas de acordo com os requisitos de processamento;
- Eficiência dos recursos computacionais.

Com o modelo físico é feita a implementação do Banco de Dados que envolve aspectos de software e de hardware que serão utilizados. O projeto físico, dependente de qual SGBD foi escolhido e então é especificado as estruturas de armazenamento, os índices e caminhos de acesso ao Banco de Dados (DATE, 2004).

Modelagem de Banco de Dados

O modelo de dados é uma junção de ferramentas possibilitam para descrição dos dados, sua semântica, relações e restrições de consistência.

Ele é fundamental para criação da estrutura de um Banco de dados, pois dará sua definição.

Os modelos de dados possibilitam uma forma de apresentação do projeto de banco de dados nos níveis Físico, Lógico e de Visão.

Modelagem de Banco de Dados

Sendo assim, os modelos de dados podem ser classificados em quatro categorias (HEUSER, 2004):

- **Modelo relacional**: Utiliza a estrutura de tabela para representação dos dados e suas relações.
- **Modelo de entidade/relacionamento**: descreve entidades e relacionamentos entre elas.
- **Modelo de dados baseado em objeto**: este modelo descreve os dados com extensões do modelo Entidade-Relacionamento, porém tem características de orientação a objeto como encapsulamento, métodos (funções) e identidade de objeto.
- **Modelo de dados semi-estruturado**: totalmente antagônico ao modelo de dados, permitindo a especificação dos dados que itens individuais do mesmo tipo q que possam ter diferentes conjuntos de atributos para representar dados estruturados utiliza a XML (Extensible Markup Language).

Modelo Entidade - Relacionamento (MER)

Em 1976 Peter Chen publicou um trabalho intitulado “The Entity-Relationship Model: Toward the unified view of data”, no qual definia o processo de modelagem de dados.

Este trabalho foi publicado e amplamente aceito, após divulgação passou a ser considerado o referencial definitivo para o processo de modelagem de dados (COUGO, 1997).

O modelo Entidade Relacionamento é composto por uma técnica de diagramação e de um conjunto de conceitos simples e serve como meio de representação dos próprios conceitos por ela manipulados (RAMAKRISHNAN e GEHRKE, 2002).

Modelo Entidade - Relacionamento (MER)

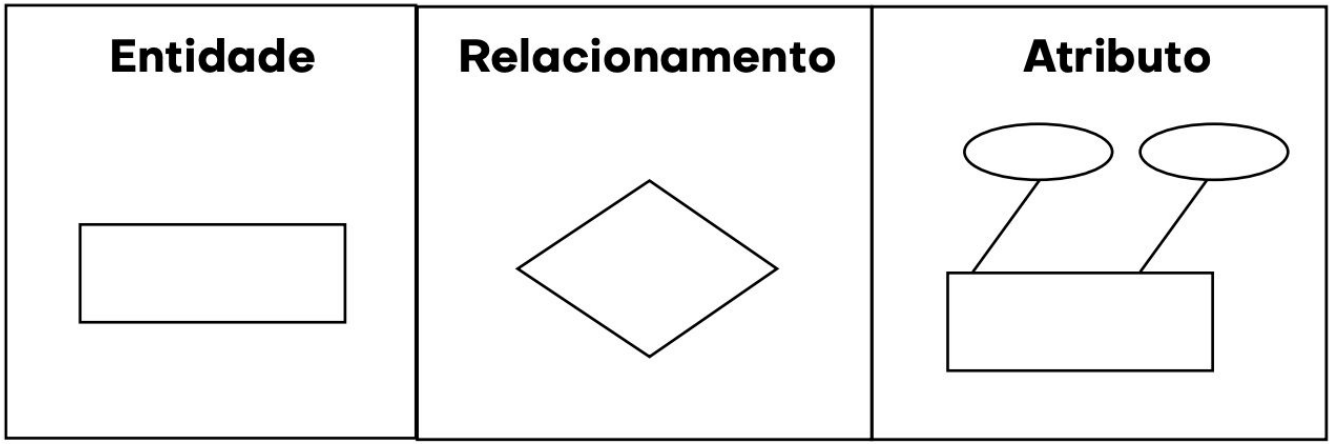
A proposta original de Peter Chen se estabeleceu e continua atualizada atualmente, e é baseada na formalização do óbvio:

“O mundo está cheio de coisas que possuem características próprias e que se relacionam entre si.”

Assim Chen visualizou um universo composto por um grupo básico de objetos chamados de entidades, com suas características próprias (os atributos) e relacionamentos entre esses objetos(entidades).

Representação Gráfica - (MER)

Podemos representar de forma gráfica o mundo real, por meio do diagrama Entidade Relacionamento para então seguir com a implementação no Banco de Dados. Para realizar a representação de cada objeto utiliza-se retângulo para representar entidades, um losango para representar os relacionamentos e elipses para indicar os atributos.



Representação Gráfica - (MER)

- Entidade - É a representação abstrata de um objeto do mundo real sobre o qual iremos guardar informações. Exemplo: Funcionários, Departamentos, Empresa, Clientes, Fornecedores, Alunos, etc. Nenhuma entende é igual a outra pois possui o conjunto de atributos diferentes.
- Atributo – São as características da entidade. Exemplo: CPF, RG, Nome, Endereço do fornecedor, Estado Civil do funcionário, Nome do aluno, Número da turma, etc.
- Relacionamento - Representa a associação (relação) entre as entidades. Por exemplo: Um aluno está matriculado em uma disciplina. Se pensarmos na instância de aluno e de disciplina, pode-se citar a situação: O João matriculado na disciplina de ginástica.

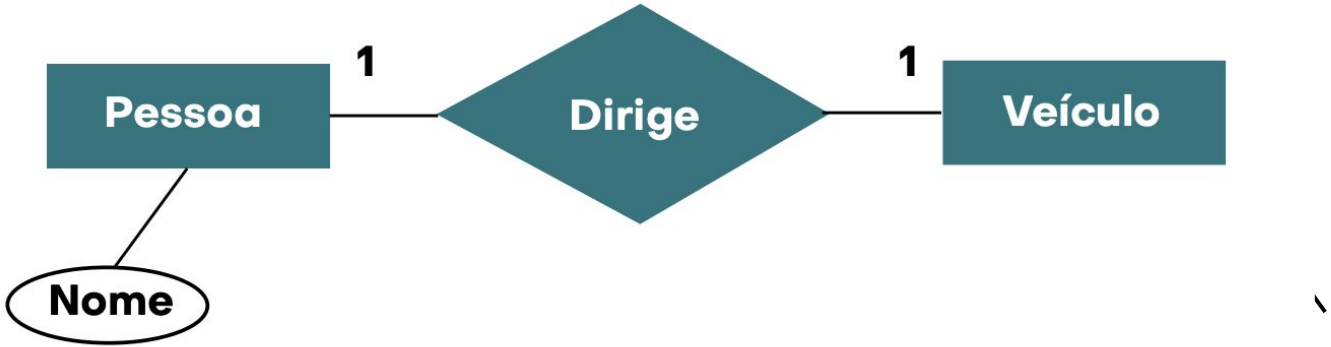
Representação Gráfica - (MER)

Exemplo:

O aluno se matriculou numa disciplina.

Representação Gráfica - (MER)

Relacionamento 1:1 - Pessoa dirige um veículo. Uma pessoa pode dirigir apenas um veículo e um veículo pode ter apenas um motorista.



Representação Gráfica - (MER)

Relacionamento 1:N ou N:1 - Empregado trabalha em departamento. Um funcionário pode trabalhar em somente um departamento e um departamento pode ter vários funcionários.

O que acontece caso inverta a cardinalidade dos relacionamentos?

Representação Gráfica - (MER)

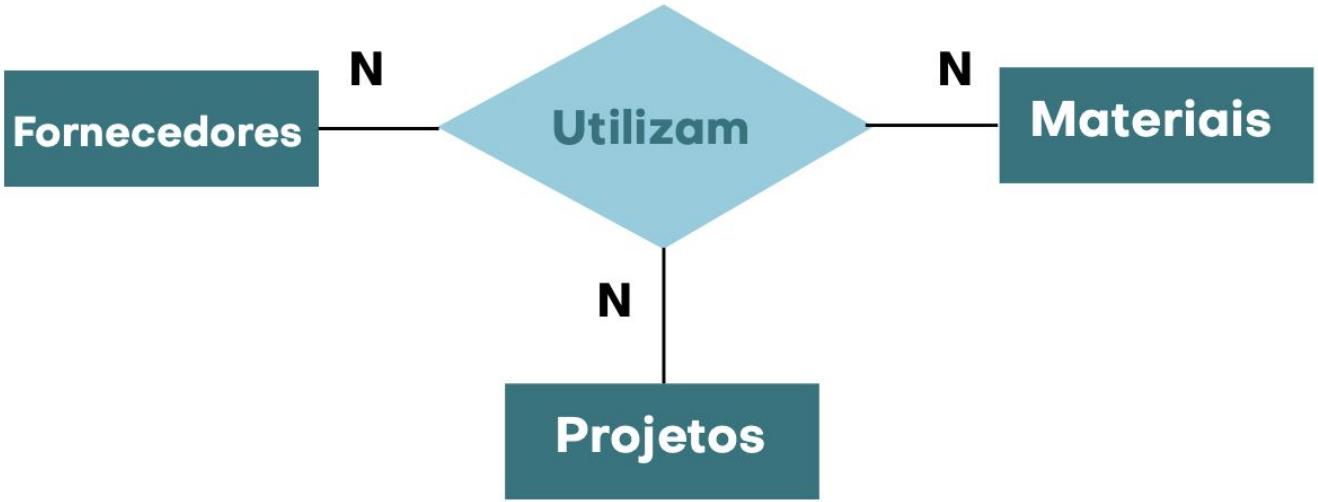
Relacionamento 1:N ou N:1 - Caso ocorra a inversão, todo o modelo conceitual será modificado. A implementação no banco seria: um empregado está lotado em VÁRIOS departamentos, mas UM departamento só pode ter um único empregado trabalhando nele.

Representação Gráfica - (MER)

Relacionamento N:M - Alunos matriculados em disciplinas. Um aluno pode estar matriculado em várias disciplinas e cada disciplina pode ter vários alunos matriculados.

Representação Gráfica - (MER)

Até agora, vimos apenas relacionamento entre duas entidades para fins de facilitação de entendimento. Mas existem maneiras de dois ou mais elementos se relacionarem. Um relacionamento pode se estabelecer entre vários elementos e não somente entre dois.



Chaves

É importante aprender o conceito de chave em modelagem de Banco de Dados, pois ela implementa restrições que garantem a integridade referencial dos dados no banco de dados.

Existem vários tipos de chave, como: chave candidata, chave primária, chave estrangeira e superchave.

Basicamente todos os atributos podem ser **candidatas a chave**. Claro, se ele tiver dados repetidos com muita frequência, não é um bom candidato à chave.

Existem alguns critérios para eleger dentre as chaves candidatas, a chave primária. **Chave primária** é o nome dado a chave candidata, escolhida por um projetista de banco de dados para identificar de forma única a entidade.

Chaves

Chaves

Chave estrangeira é o nome dado à chave primária de uma tabela que vai para outra tabela no intuito de realizar os relacionamentos entre elas. O nome estrangeira é justamente porque aquele campo não tem origem nessa tabela, sendo uma conexão entre as entidades. O campo vem de outra tabela considerada estrangeira.

Superchave é um conjunto de um ou mais atributos que, tomados coletivamente, permite-nos identificar unicamente uma entidade dentro de um conjunto de entidades. Todos os atributos que podem ser interessantes para ser chave são considerados superchave. Por exemplo, em uma tabela cliente temos o CPF, nome e data de nascimento de um cliente. Verifica-se nesse exemplo que a junção de CPF, nome cliente e data nascimento são superchave.

Representação Gráfica - (MER)

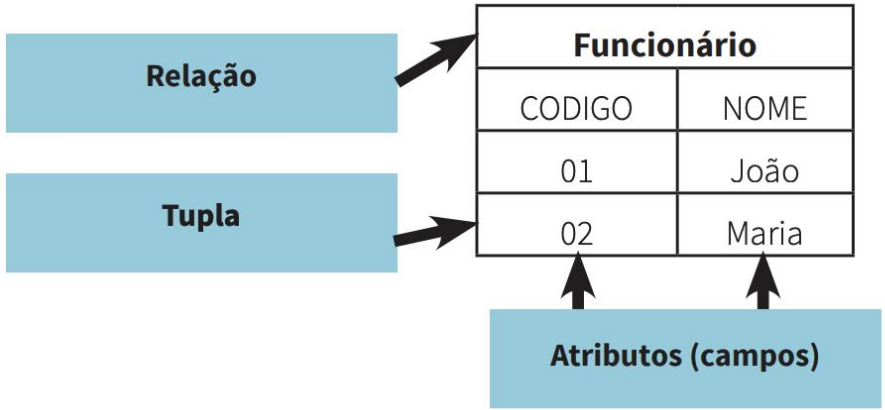
Agora que você sabe os conceitos principais do modelo Entidade Relacionamento e como fazer um diagrama, **faça os exemplos**:

- Médicos possuem uma especialidade única;
- Médicos podem ter mais de uma especialidade;
- Usuários podem ter um endereço, podem fazer mais de um pedido e cada pedido pode ter mais de um produto - MER

Lembre-se de colocar os atributos!

Teoria Relacional

A teoria relacional é usada para descrever um banco de dados por meio de conceitos simples, sendo uma forma diferente do diagrama Entidade Relacionamento (MACHADO, 2008).
É usada para **aperfeiçoar** a visão dos dados para o projetista fazendo com que a visualização do banco de dados seja vista como um conjunto de tabelas compostas por linhas e colunas.



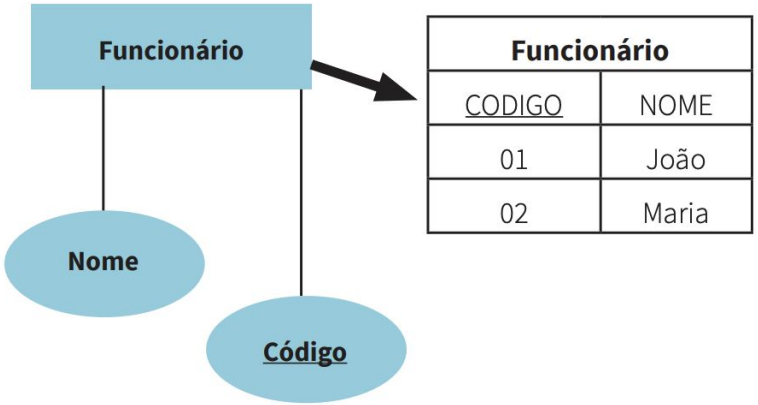
Teoria Relacional

A teoria Relacional possui premissas que definem uma tabela de dados (MACHADO, 2008):

1. Cada uma das tabelas é chamada de relação;
2. O conjunto de uma linha e suas colunas é chamado de tupla;
3. Cada coluna dessa tabela tem um nome e representa um domínio da tabela;
4. A ordem das linhas e irrelevante;
5. Não há duas linhas iguais (conceito de chave primária);
6. Usamos nomes para fazer referência às colunas;
7. A ordem das colunas também é irrelevante;
8. Cada tabela tem um nome próprio, distinto de qualquer outra tabela no banco de dados.

Teoria Relacional

Toda entidade deve ser uma tabela, e carregar todos os atributos (definidos para a entidade). Cada atributo vira um campo da tabela criada. Cada uma das chaves gera estruturas de acesso. A chave primária é projetada para não permitir ocorrências múltiplas e nem valores nulos.



Teoria Relacional

Continuaremos utilizando o exemplo que mostra a relação entre funcionário e departamento. Neste caso uma das entidades envolvidas deve carregar o identificador da outra para realizar a conexão (relacionamento) entre as tabelas, a escolha de qual tabela deverá carregar o atributo conexão deve acontecer conforme a conveniência do projeto.

Funcionário	
CODIGO	NOME
01	João
02	Maria

Departamento		
CODIGO_dept	NOME_dept	CODIGO_func
01	Contabilitade	01
02	RH	02

Teoria Relacional

Neste caso a entidade, cuja relação é do tipo várias (N), é carrega o identificador da entidade (tabela), cuja conectividade é 1 (chave estrangeira), e os atributos do relacionamento.

Funcionário		
CODIGO_func	NOME	CODdep
01	João	01
02	Maria	02

Departamento	
CODIGO_dept	NOME_dept
01	Contabilitade
02	RH

Teoria Relacional

Vejamos agora o relacionamento de muito para muitos, neste caso a relação irá torna-se uma tabela que carrega os atributos (se houver) e os identificadores das entidades que ele relaciona. Esse é o único caso em que um relacionamento torna-se uma tabela.

Funcionário	
CODIGO_func	NOME_func
01	João
02	Maria

É ALOCADO	
CODIGO_FUNC	CODIGO_PROJ
01	100
02	200

Projeto	
CODIGO_proj	NOME_proj
100	VIVA
200	NATUREZA

Banco de dados relacional x não relacional

Um banco de dados relacional é um banco de dados digital baseado no modelo relacional de dados, como proposto por E. F. Codd em 1970, uma forma intuitiva e direta de representar os dados em tabelas. As bases de dados relacionais armazenam e fornecem acesso a pontos de dados que estão relacionados entre si.

Um banco de dados não relacional é qualquer banco de dados que não segue o modelo relacional fornecido pelos sistemas tradicionais de gerenciamento de bancos de dados relacionais (SGBDR). Esta categoria de bancos de dados, é também conhecida como banco de dados NoSQL.

Banco de dados relacional x não relacional

Bancos de dados relacionais como MySQL, PostgreSQL e SQLite3 representam e armazenam dados em tabelas e filas. Bancos de dados não-relacionais como o MongoDB representam dados em coleções de documentos JSON.

As bases de dados relacionais utilizam Linguagem de Consulta Estruturada (SQL), tornando-as uma boa escolha para aplicações que envolvem o gerenciamento de várias transações. A estrutura de um banco de dados relacional permite vincular informações de diferentes tabelas através do uso de chaves (ou índices) estrangeiras. Com o SQL, você pode executar vários comandos para criar, alterar, gerenciar, consultar, dentre outras informações no seu banco de dados. Costumamos dizer que bancos SQL seguem uma modelagem relacional, pois estes se baseiam no fato de que todos seus dados sejam guardados em tabelas.

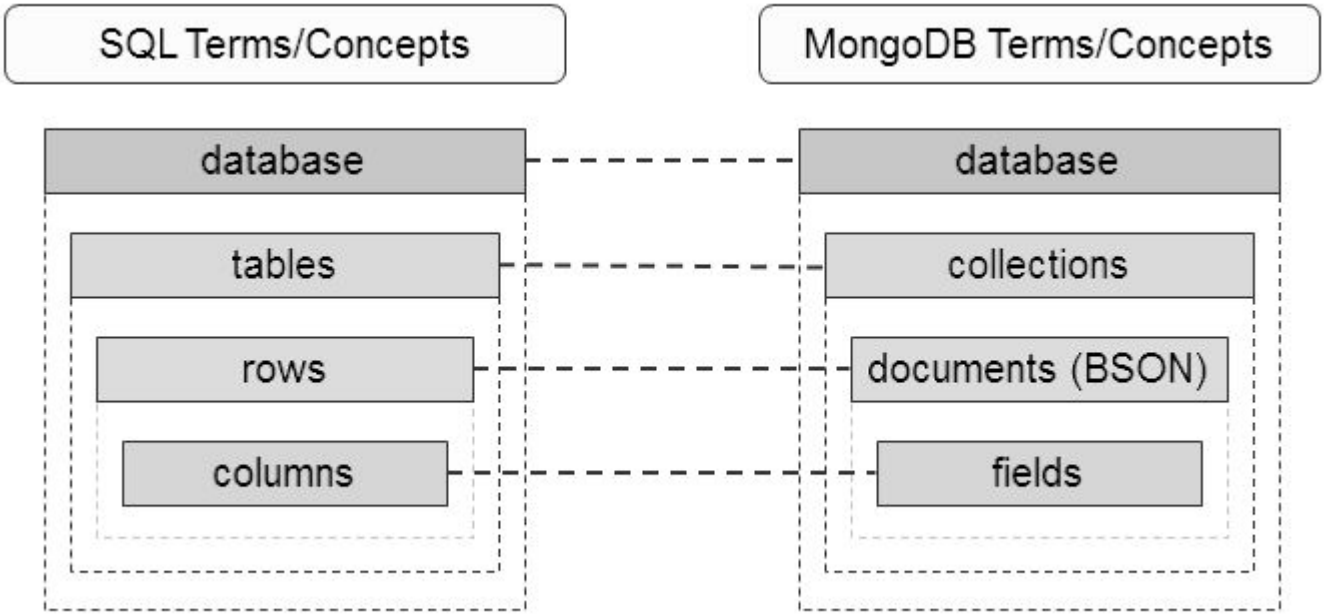
Banco de dados relacional x não relacional

Se você estiver lidando com uma **quantidade fenomenal de dados**, a **complexidade** do banco de dados relacional e das queries necessárias também **vai crescer na mesma proporção**. Nessa situação, talvez você precise considerar a possibilidade de utilizar um base de dados não relacional. Uma base de dados não relacional pode armazenar dados sem uma mecânica explícita e estruturada para vincular dados de diferentes tabelas uns aos outros.

Em bancos de dados não relacionais como MongoDB, não há joins como nos bancos de dados relacionais. Isto significa que você precisa realizar múltiplas consultas e unir os dados manualmente dentro de seu código.

Como Mongo não trata automaticamente as operações como transações da mesma forma que um banco de dados relacional, você deve escolher criar uma transação e depois verificá-la, fazer o commit ou o roll-back tudo manualmente.

Banco de dados relacional x não relacional



Banco de dados relacional x não relacional

O NoSQL tem muitas vantagens para ser utilizado. Mas não é por isso que devemos utilizá-lo em todas as situações. Em muitos sistemas, você pode (e até deve) usar o modelo relacional. O NoSQL é mais indicado para aqueles sistemas que tenham necessidades maiores de armazenamento e desempenho.

O NoSQL não veio para substituir o SQL, mas sim para oferecer mais uma alternativa de um banco de dados mais flexível no suporte de dados. Sendo assim, você pode usar ambas as soluções para diferentes casos de uso. Por isso, o mais comum em soluções escalares de sucesso é a utilização de uma arquitetura híbrida, aproveitando o melhor dois dois modelos.

MongoDB

MongoDB é um banco de dados NoSQL líder e um banco de dados de documentos de código aberto.

Com o passar dos anos, o MongoDB se tornou uma escolha popular de banco de dados altamente escalável e atualmente está sendo usado como armazenamento de dados de back-end de muitas organizações conhecidas como IBM, Twitter, Zendesk, Forbes, Facebook, Google e um zilhão de outras.

O MongoDB também chamou a atenção da comunidade de código aberto e muitos desenvolvedores trabalham em vários projetos de código aberto baseados no MongoDB. Ele é escrito principalmente em C ++.

MongoDB

O MongoDB funciona em várias terminologias que são as seguintes:

1. **Banco de dados**: - Um único servidor MongoDB consiste em vários bancos de dados, onde cada banco de dados é um contêiner físico de coleções.
2. **Coleção**: - Uma coleção no MongoDB é equivalente a uma tabela de banco de dados e existe em um único banco de dados. Inclui um grupo de documentos MongoDB.
3. **Documento**: - O documento pode ser definido como uma instância de uma coleção MongoDB. Inclui um conjunto de pares de valores-chave. Todos os documentos incluem um esquema dinâmico, o que significa que os documentos que fazem parte da mesma coleção não precisam ter o mesmo conjunto de campos e estrutura.

MongoDB

Confira abaixo um exemplo de uma collection (“tabela”) do MongoDB:


```
[{
  "_id": ObjectId("5e6261a1df9bcf90c29726d4"),
  "nome": "Henrique Marques Fernandes",
  "idade": 29
  },
  {
  "_id": ObjectId("5e6261a1df9bcf90c29726d3"),
  "nome": "Terry Crews",
  "idade": 65,
  "pais": "USA"
}]
```

MongoDB

Agora que tivemos uma visão geral sobre o MongoDB. Também é importante observar os prós e os contras do MongoDB.

Prós:
1. MongoDB é um banco de dados sem esquema. Ele não segue um projeto de esquema típico como no sistema de gerenciamento de banco de dados relacional, onde mostra uma série de tabelas e relacionamentos entre essas tabelas. No MongoDB, não há conceito de relacionamento.
2. O MongoDB é fácil de escalar.
3. Ele usa memória interna que permite um acesso mais rápido aos dados.
4. A estrutura do objeto projetado é cristalina.

MongoDB

Contras:
1. Ele fornece menos flexibilidade com consultas.
2. Sem suporte para transações.
3. Nenhuma estrutura de mesa projetada. A lógica está associada ao documento no formato JSON.

Onde usar o MongoDB?
1. Big data
2. Desenvolvimento de infraestrutura móvel.
3. Hub de dados e gerenciamento de dados do usuário.

OBRIGADA!