

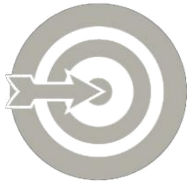


Curso de Operador de Tecnologia da Informação e Comunicações

Banco de dados

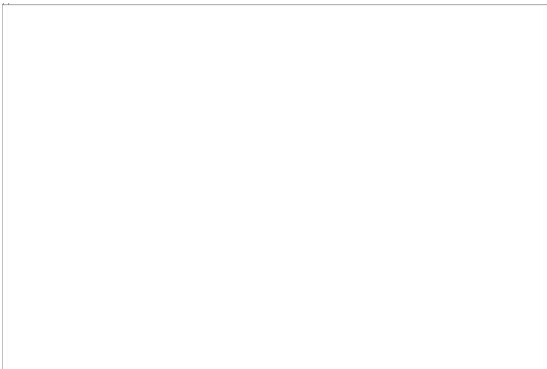
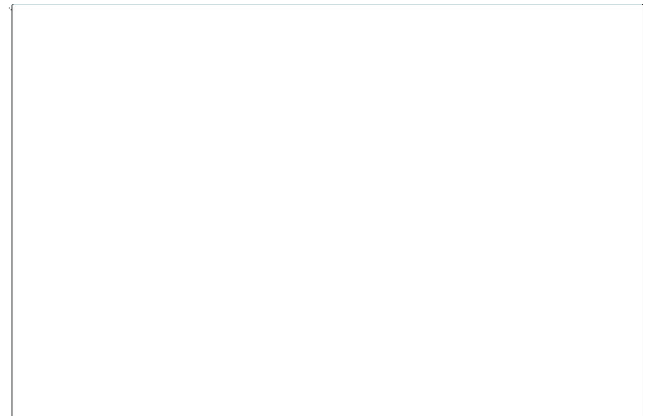
Fundamentos Básicos

Definição de Banco de dados e SGDB



Objetivo de Aprendizado

Entender os principais conceitos relacionados a banco de dados tais como: definição de banco de dados, gerenciador de banco de dados e modelagem.



Uma das principais evoluções ocorridas com os bancos de dados foi a capacidade de extração de informações que foi viabilizada pelo conceito de banco de dados relacional e a utilização de gerenciadores de banco de dados. Com essa ferramenta e as modificações que foram feitas os bancos de dados tornaram-se uma matéria essencial para todas as empresas e também no meio científico, tornando possível o avanço tecnológico e a geração de informações.

Definição de banco de dados



Existem vários tipos de banco de dados e eles estão presentes na nossa vida há muito tempo, a lista telefônica por exemplo, pode ser considerada um banco de dados.

Antigamente as empresas armazenavam informações em arquivos físicos, mas o surgimento e evolução dos computadores possibilitaram o armazenamento de dados de modo digital. Assim os bancos de dados evoluíram e se tornaram o coração de muitos sistemas de informação. A definição de Banco de dados encontrada na internet é essa:

“Bancos de dados, ou bases de dados (em Portugal), são coleções de informações que se relacionam de forma que crie um sentido. São de vital importância para empresas, e há duas décadas se tornaram a principal peça dos sistemas de informação.” Fonte: Wikipedia.

Podemos simplificar essa definição para: "**Bancos de dados** são coleções de dados interligados entre si e organizados para fornecer informações."

DADOS VS INFORMAÇÕES



Muitos consideram *dados* e *informações* como palavras sinônimas, mas na verdade não são. Para entender o que é um banco de dados é muito importante saber a diferença entre essas duas palavras.

- Dados** são fatos brutos, em sua forma primária. E muitas vezes os dados podem não fazer sentido sozinhos.

- Informações** consiste no agrupamento de dados de forma organizada para fazer sentido, gerar conhecimento.

Por exemplo, o número 2001 isoladamente faz algum sentido? Não! Isso é um dado. E se eu dissesse: "Ano do atentado terrorista às torres gêmeas: 2001"? Agora faz sentido! Isso é uma informação. **Um banco de dados é uma estrutura de dados organizada que permite a extração de informações.**

METADADOS

Todo dado relativo a outro dado é chamado de **metadados**. No exemplo anterior do ano 2001 o dado "Ano do atentado terrorista às torres gêmeas" é um metadado, pois ele é um dado sobre o dado "2001". Um outro exemplo mais comum em banco de dados: o campo telefone da tabela cliente tem, entre outros, os seguintes metadados: nome (telefone), tipo (texto), tamanho (30 caracteres), obrigatoriedade (não), etc.

Um banco de dados é formado por dados e metadados. Sem os metadados não seria possível organizar e extrair informações de um banco de dados.

Introdução aos SGBD

Um sistema de gerenciamento de banco de dados SGBD consiste em uma coleção de dados inter-relacionados e em um conjunto de programas para acessá-los. Um conjunto de dados, normalmente referenciado como banco de dados, contém informações sobre uma empresa particular, por exemplo. O principal objetivo de um SGBD é prover um ambiente que seja adequado e eficiente para **recuperar** e **armazenar informações** de banco de dados.

Os sistemas de banco de dados são projetados para gerenciar grandes grupos de informações. O gerenciamento de dados envolve a definição de estruturas para armazenamento de informação e o fornecimento de **mecanismos para manipulá-las**. Além disso, o sistema de banco de dados precisa fornecer **segurança** das informações armazenadas, caso o sistema dê problema, ou contra tentativas de acesso não-autorizado. Se os dados devem ser divididos entre diversos usuários, o sistema precisa evitar possíveis resultados anômalos.

A importância das informações na maioria das organizações e o conseqüente valor dos bancos de dados têm orientado o desenvolvimento de um grande corpo de conceitos e técnicas para o gerenciamento eficiente dos dados.



Cenário de problemas

Imaginemos parte de um banco que mantenha informações sobre todos os clientes e contas de poupança em sistemas de arquivos permanentes no banco. O sistema tem ainda uma quantidade de programas aplicativos que permitem ao usuário manipular os arquivos, incluindo:

- Um programa para debitar ou creditar em uma nova conta;
- Um programa para adicionar uma nova conta;
- Um programa para calcular o saldo de uma conta;
- Um programa para gerar posicionamentos mensais.

Esses programas aplicativos foram desenvolvidos por programadores em resposta às necessidades da organização bancária. Novos programas de aplicação são adicionados ao sistema à medida que as necessidades aparecem. Logo, mais arquivos e mais programas são acrescentados ao sistema.

O típico sistema de processamento de arquivos acima é suportado por um sistema operacional convencional. Registros permanentes são guardados em diversos arquivos, e em uma série de diferentes programas aplicativos é escrita para extrair e adicionar registros aos arquivos apropriados. Este esquema tem uma série de desvantagens:

- Redundância e inconsistência de dados:** muitos programadores diferentes e programas implementados em linguagens diferentes podem gerar arquivos de formatos diferentes. Informações podem estar duplicadas em diversos lugares. Gera inconsistência pois estas cópias podem estar com valores diferentes.
- Dificuldade no acesso aos dados:** um diretor deseja a lista de todos os clientes que moram na cidade de CEP 34863. Ou ele extrai manualmente esta informação de uma lista de clientes ou pede a um programador escrever um programa que possa fazer isso. Suponha mais tarde que o mesmo diretor deseje uma lista com os clientes com mais de \$10000. Tal lista não existe e novamente o diretor tem as duas opções.
- Isolamento dos dados:** como os dados estão espalhados, em arquivos separados e com formatos diferentes, é difícil escrever novos programas aplicativos para recuperar os dados adequados.
- Anomalias de acesso concorrente:** para aperfeiçoar o desempenho geral do sistema e obter tempo de resposta mais rápido, deixamos que múltiplos clientes acessem e atualizem os dados simultaneamente. Isso gera dados inconsistentes. Exemplo: dois clientes sacarem dinheiro de uma mesma conta corrente.
- Problemas de segurança:** nem todo usuário do sistema de BD deve ter acesso a todos os dados. Por exemplo: o RH pode ter acesso às informações cadastrais dos clientes, mas não aos valores de

conta corrente. Se novos programas aplicativos forem adicionados, é difícil assegurar tais restrições de segurança.

• **Problemas de integridade:** os valores dos dados armazenados necessitam satisfazer certas restrições. Por exemplo, o saldo nunca estar abaixo de \$25. Estas restrições podem estar contidas nos programas aplicativos, mas quando novas restrições forem adicionadas, é difícil de alterar estes programas.

Por estes motivos faz-se necessária uma abordagem relacionada à aplicação de um **SGBD**, a fim de evitar os problemas descritos.

Função de um SGBD

Estão listadas abaixo as principais funções de um gerenciador de banco de dados:

- **Segurança:** O SGBD cria um sistema de segurança que protege a base de dados de acessos não autorizados. São impostas regras que definem quais os utilizadores que podem ter acesso à base de dados, e dentro dos utilizadores autorizados a que ficheiros podem aceder e que tipos de operações podem efetuar (ler, adicionar, atualizar, apagar, etc.). Existem ainda procedimentos que permitem efetuar cópias de segurança e recuperação de dados em caso de falhas, de modo a assegurar a segurança e a integridade dos dados.
- **Segurança física:** A informação não pode estar, ao alcance de pessoas não autorizadas, ou seja, tem de estar bem guardada.
- **Segurança lógica:** A informação pode ser protegida através de métodos lógicos de segurança, exemplo passwords.
- **Integridade:** O SGBD assegura a verificação das restrições de integridade de forma a manter sempre válidos os dados, diminuindo a redundância e maximizando a consistência dos dados. A gestão de transações é um aspecto muito importante da manutenção da integridade de dados. Uma transação consiste num conjunto de ações efetuadas por um utilizador ou aplicação. Um exemplo de uma transação poderá ser uma operação de transferência de dinheiro entre duas contas. Se a transação é interrompida antes do fim (falha de energia, problemas no disco, etc.) o sistema terá de evitar um estado de inconsistência, acionando o *rollback*, que é um mecanismo que desfaz o que foi feito até à altura do problema e devolve a base de dados ao seu estado de consistência.

Além dessas funções podemos ainda destacar o **controle de concorrência**. Nas bases de dados multi-utilizadores (que serão a maioria), pode haver uma tentativa de acesso aos mesmos dados por mais de que um utilizador em simultâneo. O SGBD fornece mecanismo que asseguram que a base de dados é atualizada corretamente, este mecanismo recebe o nome de mecanismo de controlo de concorrência.

- **Backup:** Cópia de segurança, geralmente mantida em disquetes, fitas magnéticas ou CD-R, que permitem a recuperação de informações importantes ou programas em caso de falha do disco rígido.
- **Inside Backup** a informação é guardada dentro da empresa.

- **Outside Backup** a informação é guardada fora da empresa

No armazenamento de um dado, é necessário criar tabelas, dentro das quais são criadas colunas, onde serão guardadas as informações. Para que os dados presentes na base de dados fiquem bem organizados, as tabelas devem ser criadas para que não misturem as informações.

Porque relacional é melhor ?

Hoje em dia, os novos sistemas de base de dados **são quase exclusivamente do tipo relacional**.

Databases relacionais substituíram os modelos anteriores porque esses sistemas têm valiosos atributos que a distinguem como superior. Provavelmente o mais importante é **você pode mudar a estrutura de dados sem alterações nas aplicações**. Suponha, por exemplo, que você adicione uma ou mais colunas numa tabela. Você não precisa alterar nenhum aplicativo que o sistema vai continuar a processar. Claro, se você remover uma coluna que uma aplicação existente utiliza, você vai ter problemas.

Em aplicações escritas no modelo hierárquico ou rede, a estrutura de dados é praticamente embarcada na aplicação, isto é, a aplicação depende da execução física específica da Database, se você adicionar um novo atributo para o banco de dados, você deve alterar a sua aplicação para acomodar essa mudança, com ou sem a utilização do novo atributo.

Modelos relacionais oferecem flexibilidade estrutural, as aplicações escritas para esse modelo são mais fáceis de serem modificadas.

Objetos de banco de dados relacional

Veremos agora uma breve descrição dos objetos que compõem um banco de dados do tipo relacional. Eles serão vistos com mais detalhes a medida em que se aprofundaremos no curso.

Tabelas	São os objetos que contém os tipos de dados e os dados reais
Colunas ou Campos	São as partes das tabelas que armazenam os dados. Devem receber um tipo de dados e ter um nome único
Tipos de dados	Há vários tipos de dados para serem utilizados como: carácter, número, data. Um único tipo de dados é atribuído a uma coluna dentro de uma tabela
Stores Procedures (procedimentos armazenados)	São como macros em que o código Transact-SQL pode ser escrito e armazenado sob um nome.

Triggers (gatilhos) São como stored procedures que são automaticamente ativados quando os dados são inseridos, alterados ou apagados. Asseguram que regras de negócio e de integridade sejam impostas ao banco de dados.

Regras (rules) São atribuídas a colunas de modo que os dados que estão sendo inseridos devem se adaptar aos padrões definidos. Por exemplo, pode-se utilizar regras para permitir que um campo que irá armazenar a UF contenha somente Estados válidos.

Chaves Primárias (PK) Embora não sejam objetos em si, as chaves são essenciais para os bancos de dados relacionais. Promove a característica de unicidade das linhas, proporcionando uma maneira de identificar de forma única cada item que você queira armazenar.

Chaves Estrangeiras (FK) Novamente, não são objetos em si, as chaves estrangeiras são colunas que fazem referências as chaves primárias de outras tabelas.

Padrões (Defaults) Podem ser configurados em campos de modo que, se nenhum dado for inserido durante uma operação de Insert, os valores padrão serão utilizados.

Views (visualizações) Consistem basicamente em consultas armazenadas nos bancos de dados que podem fazer referência a uma ou muitas tabelas. Você pode criar e salvar views e utilizá-las no futuro. Normalmente excluem certas colunas de uma tabela e vinculam duas ou mais tabelas entre si. Podem ser utilizadas também como mecanismo de segurança.

Índices Podem ajudar os dados de modo que as consultas executem mais rápido

Principais SGBD do mercado

Oracle



Originada nos anos 80, criada por Larry Ellison, a Oracle é hoje uma das maiores empresas de tecnologia do mundo, sendo que desde 2009 é também proprietária das linhas de software Java.

O SGBD é o seu principal produto e desde o seu lançamento no mercado vem sendo aperfeiçoado e desenvolvido para atender as necessidades de empresas e do mundo conectado em que vivemos. Existem diversas versões do software, sendo que cada uma delas contam com características que a tornam ideal à diferentes modelos de negócio. É um software focado para as necessidades de empresas de médio e grandes porte.

Vale lembrar que, até o momento, o **SGBD da Oracle é relacional**. Para manipulação e gestão do sistema, utiliza-se a linguagem PL/SQL. Linguagem desenvolvida pela Oracle, a partir do SQL (ANSI), expandindo a sua capacidade original.

Com amplo domínio de mercado, conhecer a linguagem utilizada no SGBD da Oracle é fundamental para o profissional de TI que visa ingressar na carreira como um DBA. Você pode ter mais informações acessando o site oficial da Oracle. No Brasil, possuímos poucas empresas consolidadas na prática do ensino a administração do SGBD da Oracle, contudo, a TargetTrust destaca-se nesse segmento

MySQL



Um dos bancos de dados mais utilizados e populares, trata-se de uma tecnologia **Open Source**, ou seja, de **código aberto**. Isto permite que o desenvolvimento do mesmo seja realizado de acordo com as necessidades de uma organização.

Além disso, sua facilidade de uso e capacidade de rodar em diferentes sistemas operacionais garante lugar cativo na preferência de grandes empresas por todo o globo. É considerado o SGBD mais popular no mundo.

Oracle + MySQL = ?

Atualmente, pertence a Oracle que, além de fomentar o uso da tecnologia Open Source, ainda disponibiliza uma versão mais robusta (paga, é claro), contando com funcionalidades da empresa, como: alta disponibilidade de monitoramento, firewalls, backups avançados, auditoria e entre outras.

Para o profissional de TI, **conhecer e saber utilizar o MySQL**, que é SGBD relacional, pode ser considerado tão importante quanto o conhecimento do SGBD Oracle. Em outras palavras, é **praticamente obrigatório** para quem deseja aproveitar a todas as oportunidades do mercado (empresas de pequeno, médio e grande porte).

SQL Server



O poderoso **SGBD relacional da Microsoft**, lançado em 1988 como parte do Windows NT e posteriormente comercializado como um produto separado e em constante desenvolvimento desde então.

Seu grande diferencial com relação às outras opções já citadas é a possibilidade do desenvolvedor utilizar linguagens de programação gerenciadas, como o C# e o Visual Basic .NET, ao invés de usar declarações SQL. Obviamente, você também pode utilizar o SQL, ou melhor, o T-SQL – extensão do SQL (ANSI) para o MS SQL Server. O MS SQL Server também possibilita consultas transparentes.

O Microsoft SQL Server é uma opção extremamente poderosa e, apesar de ser uma solução totalmente paga, sempre figura entre os TOP 5 dos principais rankings da área.

PostgreSQL



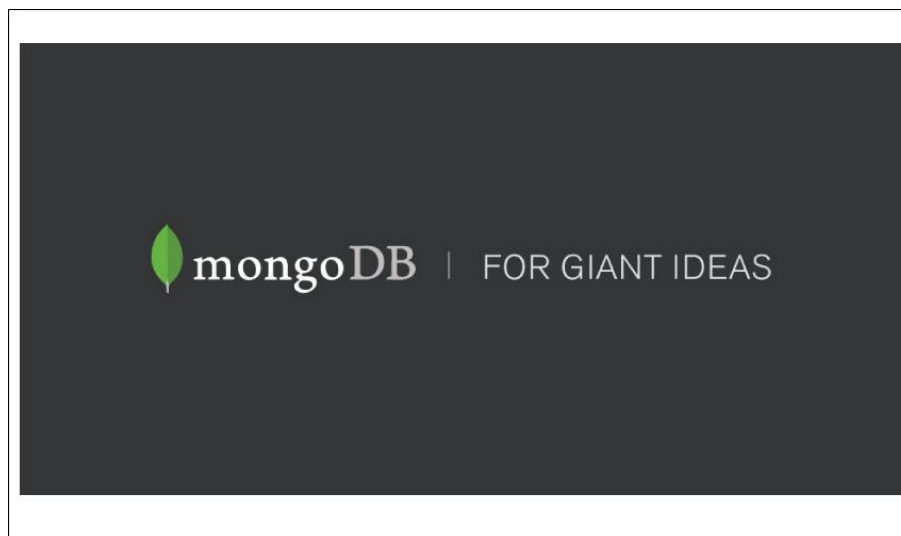
Banco de dados relacional e **Open Source**, desenvolvido pela **PostgreSQL Global Development Group**. Lançado em 1989 e frequentemente se estabelece entre os cinco sistemas de gerenciamento de bancos de dados mais utilizados por organizações do mundo todo.

Assim como MySQL, é muito utilizado para sistemas web, permitindo que o desenvolvimento de soluções seja realizado de forma que atenda às necessidades específicas de um negócio, desta forma, alcançando um melhor desempenho.

The world's most advanced open source database (Em português, "O banco de dados open source mais avançado do mundo").

Este é o slogan do PostgreSQL. No Brasil, organizações públicas tem procurado se especializar na tecnologia, visando a adoção do software livre, bem como, otimização e organização dos dados e, claro, redução de custos!

MongoDB



Um dos SGBDs que mais cresceu nos últimos anos. Trata-se de um **sistema NoSQL** que foi lançado em 2009 pela empresa que leva o mesmo nome.

O SGBD procura unir o melhor dos sistemas relacionais e as inovações do NoSQL, mantendo muitas características do primeiro, como índices e consultas dinâmicas, mas também com o modelo de dados orientados a documentos.

Dessa forma, os ganhos de agilidade através de esquemas flexíveis e a maior facilidade na escalabilidade horizontal, são relevantes quando comparados a outras soluções. O **software é Open Source**, assim como, o MySQL e o PostgreSQL. O que permite o desenvolvimento do sistema conforme as necessidades do negócio.

Com o crescimento da preferência no uso do Mongo DB, o profissional de TI que dominar **JavaScript (linguagem utilizada pelo sistema)**, possui grandes chances de se destacar no mercado atual.

DB2



Banco de dados da gigante IBM. Lançando ainda na década de 1980, embora seu desenvolvimento tenha ocorrido durante os anos de 70. Se valendo da teoria dos bancos de dados relacionais, desenvolvida por Edgar Frank Codd, que na época trabalhava para a empresa.

A IBM não acreditava no potencial de Codd e na linguagem que havia desenvolvido, deixando-o de fora de seus projetos e menosprezando diversas ideias do matemático. Em outras palavras, o **pai da teoria dos bancos de dados relacionais** foi um mero figurante no desenvolvimento do produto final. Cruel, não?

Voltando ao **DB2 e a IBM**, o sistema é marcado pelo pioneirismo e, além disso, pelo seu suporte aos sistemas operacionais.

Historicamente, disputou a primeira posição do mercado com a Oracle durante muitos anos. Contudo, o surgimento e desenvolvimento de novas funcionalidades nos outros SGBDs contribuíram para uma significativa perda de espaço para o DB2. Ainda é um sistema extremamente relevante.

SQLite



Desenvolvido por Richard Hipp, o SQLite é uma **biblioteca em linguagem C**, que permite o uso de um banco de dados SQL. Sendo que, ao invés de se conectar a um grande servidor de banco de dados, ele é o próprio servidor.

Seu uso é recomendado quando há facilidade na administração e a implementação e manutenção dos dados são fatores mais importantes que os incontáveis recursos que os demais SGBDs podem oferecer. É ideal para sites com poucos acessos, aplicações desktop, ferramentas estáticas de análise e entre outras.

Além disso, é um **software livre** e de domínio público, suporta dados abaixo de 2 terabytes e não tem dependências externas, o que facilita seu uso por pequenas e médias empresas.



Todo agrupamento de bancos de dados **possui um conjunto de usuários** de banco de dados. Estes usuários **são distintos dos usuários gerenciados pelo sistema operacional** onde o servidor executa. Os usuários possuem objetos de banco de dados (por exemplo, tabelas), e podem **conceder privilégios** nestes objetos para outros usuários controlando, assim, quem pode acessar qual objeto.

Administrador de Banco de Dados (DBA)

Em um ambiente de banco de dados, o recurso primário é o banco de dados por si só e o recurso secundário o SGBD e os softwares relacionados. A administração destes recursos cabe ao **Administrador de Banco de Dados**, o qual é **responsável pela autorização de acesso** ao banco de dados e pela **coordenação e monitoração de seu uso**. Ou seja, ele coordena todas as atividades do sistema de banco de dados; possui boa compreensão dos recursos de informação da empresa e suas necessidades.

Suas funções incluem:

- Definição do esquema
- Estrutura de armazenamento e definição de acesso aos dados
- Esquema físico e organização
- Concede acesso aos usuários
- Cuida da integridade dos dados
- Atua como elo com os usuários
- Acompanha a desempenho, e responde as mudanças exigidas
- Atividades de manutenção (Backups)

Projetista de Banco de Dados

O Projetista de Banco de Dados é **responsável pela identificação dos dados que devem ser armazenados** no banco de dados, escolhendo a estrutura correta para representar e armazenar dados. Muitas vezes, os projetistas de banco de dados atuam como "staff" do DBA, assumindo outras responsabilidades após a construção do banco de dados. É função do projetista também **avaliar as necessidades de cada grupo de usuários para definir as visões** que serão necessárias, integrando-as, fazendo com que o banco de dados seja capaz de atender a todas as necessidades dos usuários.

Usuários Finais

Existem basicamente **três categorias de usuários finais** que são os usuários finais do banco de dados, fazendo consultas, atualizações e gerando documentos:

- Usuários casuais**: acessam o banco de dados casualmente, mas que podem necessitar de diferentes informações a cada acesso; utilizam sofisticadas linguagens de consulta para especificar suas necessidades;

- **Usuários novatos** ou paramétricos: utilizam porções pré-definidas do banco de dados, utilizando consultas pre estabelecidas que já foram exaustivamente testadas;
- **Usuários sofisticados**: são usuários que estão familiarizados com o SGBD e realizam consultas complexas.



Modelar significa criar um modelo que explique as características de funcionamento e comportamento de um software a partir do qual ele será criado, facilitando seu entendimento e seu projeto, através das características principais que evitarão erros de programação, projeto e funcionamento. É uma parte importante do desenho de um sistema de informação.(Wikipédia)

Os modelos de dados são ferramentas que permitem demonstrar como serão construídas as estruturas de dados que darão suporte aos processos de negócio, como esses dados estarão organizados e quais os relacionamentos que pretendemos estabelecer entre eles.

A abordagem que se dispensa ao assunto normalmente atende a três perspectivas:

- **Modelagem Conceitual:** é usada como representação de alto nível e considera exclusivamente o ponto de vista do usuário criador dos dados. O modelo conceitual é um diagrama em blocos que demonstra todas as relações entre as entidades, suas especializações, seus atributos e auto-relações.;
- **Modelagem Lógica:** agrega mais alguns detalhes de implementação.O modelo lógico mostra as ligações entre as tabelas de banco de dados, as chaves primárias, os componentes de cada uma, etc.
- **Modelagem Física:** demonstra como os dados são fisicamente armazenados. Inclui a análise das características e recursos necessários para armazenamento e manipulação das estruturas de dados (estrutura de armazenamento, endereçamento, acesso e alocação física), sendo uma sequência de comandos executados em SQL a fim de criar as tabelas, estruturas e ligações projetadas até então e finalmente criar o banco de dados.

AS ETAPAS DA ELABORAÇÃO DE UM PROJETO DE BANCO DE DADOS

Para o sucesso de um projeto, é preciso seguir alguns passos. Vamos conhecer cada um deles.

1 - Análise de requisitos

O primeiro passo para a criação do modelo correto é entender a necessidade. A isso chamamos de Análise de Requisitos.

É preciso, primeiramente, levantar os requisitos que darão a ideia inicial do sistema. Nessa fase — que é considerada a mais importante do processo —, procura-se entender a regra de negócio estabelecida, por meio da qual será desenhado o que chamamos de minimundo. Esse termo procura descrever o universo de relações de negócios que serão atendidas pelo modelo a ser implementado.

Na análise de requisitos, atividades como entrevistas e reuniões para discutir os processos que serão controlados pelo sistema são de suma importância para que o desenho do modelo seja o mais fiel possível à realidade e atenda de forma satisfatória à expectativa do interessado.

Pode-se dizer, sem medo, que o sucesso das próximas fases dependerá do entendimento claro da necessidade do cliente.

2 - Modelo conceitual

A seguir há o Modelo Conceitual. Nessa etapa as informações obtidas no passo anterior servem como entrada para geração do desenho conceitual. Costuma-se utilizar o recurso chamado modelo entidade-relacionamento, em que se define os agentes que farão parte das rotinas e os relacionamentos entre esses agentes.

Por exemplo, há uma entidade chamada “cliente” e outra chamada “carro”. O relacionamento que é estabelecido entre essas duas entidades provavelmente será chamado de “compra”. Assim, no desenho conceitual, tem-se que “o cliente compra o carro”.

Nesse momento, também se estabelece a cardinalidade entre as entidades, ou seja, usando o exemplo citado, definem-se as regras, como:

- Se um cliente compra um ou vários carros;
- Se um carro pode ser comprado por um ou vários clientes.

Nesse ponto, já se inicia a materialização das regras de negócios discutidas na primeira etapa. Se fosse uma locadora de automóveis, um carro poderia ser alugado por um ou mais clientes (claro, apenas por um de cada vez).

Já no caso da loja, o carro só pode ser vendido uma vez para um cliente. A não ser que seja uma loja de carros usados ou seminovos: nesse caso, existe a possibilidade de o cliente voltar algum tempo depois e trocar o carro por outro mais novo. Tudo isso é definido na análise de requisitos.

3 - Modelo lógico

Nessa etapa, são definidos os aspectos físicos do projeto de banco de dados. Cada entidade tende a se tornar uma tabela, que armazenará informações. Nessa fase é hora de definir os tipos de dados e tamanho dos campos que vão armazenar as informações. Por exemplo:

- RG e CPF pedem dados do tipo numérico;
- Nomes, endereços e informações que contenham tanto letras como números são dados do tipo caractere.

Também será determinado o tamanho dos campos. Essa fase é preponderante para o sucesso do modelo, pois um dimensionamento errado vai redundar em retrabalho para correção futura.

Geralmente, o custo maior em desenvolvimento de sistemas é exatamente para correção e ajustes posteriores à entrega do projeto. Por isso, essa fase é fundamental para o sucesso de um projeto de banco de dados.

4 - Modelo físico

Aqui, chega a última, mas não menos importante fase do desenho. Nesse momento já existe uma ideia bastante precisa de banco de dados. Nessa fase é que os detalhes físicos são definidos: armazenamento dos dados, geração dos scripts que irão criar as tabelas e visões do banco.

Essa fase também marca um importante detalhe: todas as fases anteriores foram executadas independente da plataforma de software de banco de dados que seria usada. Agora, cada ação está diretamente ligada à solução de SGDB (Sistema Gerenciador de Banco de Dados). Para cada produto existente no mercado, há as melhores práticas e recomendações dos especialistas em administração de bancos de dados, também chamados DBAs.

4.1 A sintaxe dos scripts

Aqui, se define a sintaxe dos scripts para o produto específico: Oracle, SQL Server, MySQL ou PostgreSQL, por exemplo. Cada um tem suas especificações de instalação e melhores práticas, bem como plataformas onde serão executados.

Por exemplo, o Oracle pode ser usado em vários sistemas operacionais, como Windows, Linux e Solaris. Já o SQL Server é instalado no sistema operacional do seu fabricante, a Microsoft. Esses detalhes são discutidos nessa fase, levando-se em conta aspectos técnicos e orçamentários, como aquisição de licenças e especialistas no produto para suporte.

4.2 As regras de segurança

Nessa fase também se aplicam as regras de segurança: quem terá acesso a qual informação no banco de dados, bem como rotinas de backup e restauração. Soluções de administração do banco de dados, replicação ou redundância também podem ser definidos aqui.

Dessa forma, perceberemos duas peculiaridades no projeto de banco de dados:

- No início do projeto, a tônica é a necessidade do solicitante. Ouvi-lo nessa fase é de suma importância para traduzir o desenho em um modelo coerente e satisfatório.
- No final do projeto, o foco maior está em como implementar o projeto, garantindo a performance e disponibilidade do sistema.

Referência

<https://dicasdeprogramacao.com.br/o-que-e-um-banco-de-dados/>

<https://profissaodba.wordpress.com/historico-dos-bancos-de-dados/>

<https://www.devmedia.com.br/gerenciamento-de-banco-de-dados-analise-comparativa-de-sgbds/30788>

<http://ehgomes.com.br/disciplinas/bdd/sqbd.php>

<https://becode.com.br/principais-sgbds/>

https://pt.wikipedia.org/wiki/Modelagem_de_dados

<https://www.impacta.com.br/blog/2017/02/10/veja-como-elaborar-um-projeto-de-banco-de-dados/>