FACULDADES FACCAT

SISTEMAS DE INFORMAÇÃO

ADRIAN HIDEKI DOS SANTOS

**SOFTWARE DE OTIMIZAÇÃO DE BANCO DE DADOS**

**TUPÃ**

**2020**

ADRIAN HIDEKI DOS SANTOS

SOFTWARE DE OTIMIZAÇÃO DE BANCO DE DADOS

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de bacharel em Sistemas de Informação.

Orientador: Ewerton Silva

Tupã

2020

**SUMÁRIO**

[1. Introdução 6](#_Toc40300829)

[1.1. Definição do problema 6](#_Toc40300830)

[1.2. Objetivos 6](#_Toc40300831)

[1.2.1. Objetivo Geral 6](#_Toc40300832)

[1.2.2. Objetivo Específico 7](#_Toc40300833)

[1.3. Justificativa 7](#_Toc40300834)

[1.4. Metodologia 7](#_Toc40300835)

[2. Fundamentação teórica 7](#_Toc40300836)

[2.2. História do banco de dados 7](#_Toc40300837)

[2.4. Normalização de dados 10](#_Toc40300838)

[2.5. SQL Server 11](#_Toc40300839)

[2.6. Recursos do SQL Server 11](#_Toc40300840)

[2.7. Padrões de desenvolvimento e melhores práticas 11](#_Toc40300841)

[2.8. Indexação de tabelas e estrutura de dados 11](#_Toc40300842)

[2.9. Tipos de dados 11](#_Toc40300843)

[2.10. Arquivos do banco de dados 11](#_Toc40300844)

[3. Desenvolvendo o projeto proposto 11](#_Toc40300845)

[3.1. Diagramas UML 11](#_Toc40300846)

[3.1.1. MER - Modelo de Entidade e Relacionamento 11](#_Toc40300847)

[3.1.2. Dicionário de dados 11](#_Toc40300848)

[3.1.3. Diagrama de Classes 11](#_Toc40300849)

[3.1.4. Descrição dos Casos de Uso 11](#_Toc40300850)

[3.2. Requisitos não funcionais 11](#_Toc40300851)

[3.2.1. Manutenção 11](#_Toc40300852)

[3.2.2. Suporte 11](#_Toc40300853)

[3.2.3. Infraestrutura 11](#_Toc40300854)

[3.2.4. Segurança 11](#_Toc40300855)

[3.3. Métodos para controle de segurança do sistema 11](#_Toc40300856)

[3.3.1. Controle de Segurança Lógica 11](#_Toc40300857)

[3.3.2. Plano de Contingência 11](#_Toc40300858)

[3.4. Layout dos Relatórios 11](#_Toc40300859)

[3.4.1. Nome do relatório 11](#_Toc40300860)

[3.4.1.1. Tela do relatório 11](#_Toc40300861)

[3.4.1.2. Exemplo do relatório 11](#_Toc40300862)

[3.4.1.3. Instruções SQL 11](#_Toc40300863)

[3.4.1.4. Ferramenta utilizada para desenvolver e apresentar o relatório 11](#_Toc40300864)

[4. Implementações Futuras 11](#_Toc40300865)

[5. Conclusão 11](#_Toc40300866)

[6. Referências Bibliográficas 11](#_Toc40300867)

[7. Relatório de Participação dos alunos no estágio 12](#_Toc40300868)

LISTA DE ILUSTRAÇÕES

[Figura 1 - Edgar Frank Codd 7](#_Toc40295738)

# Introdução

Atualmente muitas soluções possuem sua lógica de negócio completamente concentrada no banco de dados, ou seja, em regras de negócio e tecnologia, torna-se cada vez mais importante prezar pelo controle e performance do que acontece no banco de dados, para que os clientes tenham maior fluidez ao utilizar o sistema e os usuários consigam fazer as operações normalmente com eficiência e praticidade.

Com os padrões definidos pela tecnologia, de melhores práticas, seja para estruturação e codificação, torna-se importante cada vez mais adequar os programas existentes para este padrão, para sistemas antigos acaba sendo necessário revisar os códigos existentes, gerando atividades que não seriam necessárias se fosse utilizado um padrão de desenvolvimento eficiente.

O SQL Server possui uma linguagem chamada T-SQL (transact SQL), onde é possível de modo transacional, interagir com o banco de dados, fazendo consultas e atualizações no mesmo. Quando necessitamos processar as informações no banco de dados, é necessário utilizar das funções disponíveis na linguagem T-SQL, todavia deve sempre atentar-se com relação às operações no banco de dados que possuem filtros ou ordenações pois quando não existem índices e suporte uma consulta pode ficar muito lenta, consumindo recursos desnecessários do servidor.

Os índices são objetos no banco de dados criados para otimizar operações no banco de dados para determinadas tabelas e visões, basicamente os índices ordenam os registros de uma tabela, a partir dos campos, ou seja, quando um índice é criado precisamos informar as colunas chave, criando o índice quando realizarmos uma consulta utilizando de filtros os campos do índices, onde temos um ganho de performance, pois não é necessário ler toda a tabela ou ordenar os dados para realizar a pesquisa.

## Definição do problema

A lentidão no banco de dados pode ser causadas por diversos fatores, seja uma tabela muito grande, comandos utilizados de forma incorreta, a ausência de indexação ou a má indexação de uma e várias tabelas, onde podem afetar as funcionalidades do sistema tornando-as lentas impedindo o fluxo do usuário. Ainda, configurações indevidas, no banco, ou tabelas com uma grande volumetria, podem atrapalhar o funcionamento do ambiente, prejudicando todos os usuários do ambiente.

## Objetivos

### Objetivo Geral

Desenvolver um sistema de otimização de banco de dados, tornando possível gerar relatórios de pontos que possam ser melhorados no sistema, com relação ao banco de dados, sendo possível fazer análises de informações presentes na base e de rotinas sendo executadas em tempo real. A análise da indexação das tabelas e dos relacionamentos, assim como estatísticas que influenciam totalmente na performance de uma aplicação no banco de dados.

### Objetivo Específico

Identificar problemas que degradam a performance do banco de dados, criar e alterar índices em tabelas no banco de dados, de forma que facilite o trabalho de identificação de problemas e customização de índices na base de dados.

* Estudar e apresentar as melhores práticas de banco de dados;
* Desenvolver um sistema que automatize verificações realizadas manualmente;
* Exemplificar situações que podem afetar a performance de um sistema;

## Justificativa

Cada vez mais se torna necessário realizar análises nos servidores de banco de dados, devido a quantidade de itens que podem afetar o desempenho do servidor, por isso torna-se necessário economizar tempo e identificar o problema de forma automática e que o resultado seja compreendido, considerando sempre as melhores práticas de configuração e programação no banco de dados. Com isso, é possível evidenciar para os clientes os problemas que afetam o servidor e o sistema em si.

## Metodologia

Será consultada a documentação do produto SQL Server atualizada, assim como sites de profissionais nesta tecnologia para que seja sempre levando em consideração os melhores padrões de configuração e desenvolvimento. O software será desenvolvido em C#, utilizando os recursos mais atualizados do .net framework.

# Fundamentação teórica

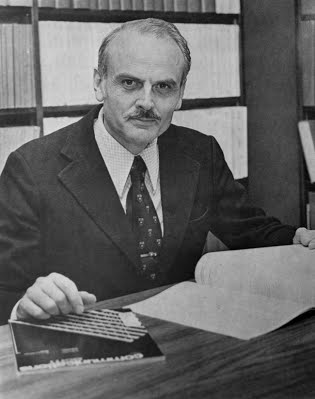
## Introdução ao banco de dados

O banco de dados é um aplicativo que armazena as dados em formato de tabelas, contendo linhas e colunas. Um banco de dados pode ter diversas tabelas, assim como uma tabela pode ter várias linhas. Sempre que criamos uma tabela, temos o objetivo de armazenar um conjunto de dados sobre uma entidade específica, ou seja, se criarmos uma tabela de pessoas, criaremos campos que armazenam dados de uma pessoa, como nome, data de nascimento, peso. Segundo ALVES (2010): “Podemos simplificar essa definição para: Bancos de dados são coleções de dados interligados entre si e organizados para fornecer informações.”.

## História do banco de dados

A história do banco de dados tem seu início com os softwares de armazenamento de informações (Clipper, Dbase 2, Fox Pro, COBOL), os mesmos disponibilizavam métodos e funcionalidades para armazenar informações e selecionar as mesmas. Durante os anos de 1970 e 1972 surgiu o modelo de banco de dados relacional, proposto por Edgar Frank Codd, isso fez com que o modo em que os dados são armazenados não estivessem relacionados com a estrutura lógica do banco de dados.

Figura 1 – Edgar Frank Codd



*Fonte: Blog poineiroscomputação.*

O banco de dados relacional melhorou muito os projetos de banco de dados, de forma que não ocorram redundâncias e campos desnecessários nas tabelas, ganhando armazenamento e melhorando a performance. Em 1976 o Dr. Peter Chen propôs o modelo de entidades e seus relacionamentos (conhecido atualmente como MER), sendo uma técnica de criação de projetos de banco de dados, capaz de descrever a estrutura inteira do banco e seus respectivos relacionamentos, sendo fácil, intuitivo e muito prático em casos de dúvidas na fase de desenvolvimento e manutenção do produto. Em 1980 segundo Rezende:

A Linguagem Estruturada de Consulta – SQL (Structured Query Language) se torna um padrão mundial. A IBM transforma o DB2 como carro chefe da empresa em produtos para BD. Os modelos em rede e hierárquico passam a ficar em segundo plano praticamente sem desenvolvimentos utilizando seus conceitos, porém vários sistemas legados continuam em uso. O desenvolvimento do IBM PC desperta muitas empresas e produtos de BD como: RIM, RBASE 5000, PARADOX, OS/2 Database Manager, Dbase III e IV (mais tarde transformado em FoxBase e mais tarde ainda como Visual FoxPro), Watcom SQL, entre outros.

Em 1990, um modelo de aplicação começou a ganhar espaço no mercado, de acordo com Rezende “O modelo cliente-servidor (client-server) passa a ser uma regra para futuras decisões de negócio e vemos o desenvolvimento de ferramentas de produtividade como Excel/Access (Microsoft) e ODBC”. Na metade dos anos 90, os processos de transação em tempo real atingem uma estabilidade com seu intenso uso em pontos de venda (REZENDE). A partir dos anos 2000, os sistemas gerenciadores de banco de dados evoluíram muito, tendo capacidade de armazenar grande quantidade de dados e aumentaram os recursos fornecidos para que se torne cada vez mais fácil trabalhar com as informações no banco de dados. Atualmente os bancos de dados seguem com a tendência de não ser somente um repositório de dados, mas sim uma fonte de informações que torna-se possível extrair estatísticas e assim fornecer uma informação que seja possível utilizar em uma tomada de decisão, temos diversos bancos de dados que suportam capacidade superiores a Terabytes de informações, e tecnologias que conseguem processar as mesmas de acordo com as necessidades de negócio e níveis estratégicos da empresa. Recentemente as empresas têm investido muito em bancos de dados potentes, apostando em sistemas relacionais e não relacionais, visando sempre o melhor desempenho e praticidade em suas aplicações.

* 1. SGBD

Os SGBD nada mais são do que os Sistemas de gerenciamento de banco de dados, os mesmos são aplicações que tem o intuito de gerenciar o banco de dados, sendo responsável por fazer os processamentos, armazenar, atualizar, apagar e selecionar informações no banco de dados que está sendo gerenciado, segundo Sanches:

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 consequente valor dos bancos de dados têm orientado o desenvolvimento de um grande corpo de conceitos e técnicas para o gerenciamento eficiente dos dados. (2005).

O sistema gerenciador do banco de dados deve ser uma aplicação completa, onde seja possível garantir a integridade dos dados e que seja possível também realizar atividades de contingência, onde caso ocorra algum imprevisto, seja possível recuperar as informações, sendo um mecanismo conhecido pelo termo de *backup*.

## Normalização de dados

A normalização do banco de dados é uma das técnicas mais importantes utilizadas na modelagem do banco de dados e na criação de suas tabelas, com a utilização da mesma, temos diversos aproveitamentos do banco de dados em si, observando o entendimento do sistema, custo de armazenamento e otimização do mesmo, segundo Machado (2015):

Normalização é o processo de modelar o banco de dados projetando a forma como as informações serão armazenadas a fim de eliminar, ou pelo menos minimizar, a redundância no banco. Tal procedimento é feito a partir da identificação de uma anomalia em uma relação, decompondo-as em relações melhor estruturadas.

Normalmente precisamos remover uma ou mais colunas da tabela, dependendo da anomalia identificada e criar uma segunda tabela, obviamente com suas próprias chaves primárias e relacionarmos a primeira com a segunda para assim tentarmos evitar a redundância de informações.

Ao se utilizar a normalização de dados em um banco de dados, temos cinco formas para aplicar e assim normalizar um banco de dados, de forma simples, o objetivo da normalização do banco de dados é eliminar a duplicidade de informações e também evitar que informações não obrigatórias sejam armazenadas na mesma tabela, analisando todas as formas normais cada forma faz com que os dados sejam mais sintéticos e íntegros. Para considerar um banco de dados como normalizado não é necessário aplicar todas as formas normais, aplicando até a terceira forma já é possível considerar como normalizada (MARCHI, 2013).

* + 1. Primeira forma

A primeira forma consiste em remover as tabelas aninhadas, ou seja, ter duas entidades na mesma tabela (MARCHI, 2013). Nesse caso é necessário separar os campos que correspondem as respectivas entidades em duas tabelas distintas, cada uma com suas devidas informações.

* + 1. Segunda forma
    2. Terceira forma
    3. Quarta forma
    4. Quinta forma

## SQL Server

## Recursos do SQL Server

## Padrões de desenvolvimento e melhores práticas

## Indexação de tabelas e estrutura de dados

## Tipos de dados

## Arquivos do banco de dados

# Desenvolvendo o projeto proposto

## Diagramas UML

### MER - Modelo de Entidade e Relacionamento

### Dicionário de dados

### Diagrama de Classes

### Descrição dos Casos de Uso

* + - 1. …
      2. ....

## Requisitos não funcionais

### Manutenção

### Suporte

### Infraestrutura

### Segurança

## Métodos para controle de segurança do sistema

### Controle de Segurança Lógica

### Plano de Contingência

## Layout dos Relatórios

### Nome do relatório

#### Tela do relatório

#### Exemplo do relatório

#### Instruções SQL

#### Ferramenta utilizada para desenvolver e apresentar o relatório

# Implementações Futuras

# Conclusão

# Referências Bibliográficas

MICROSOFT SQL SERVER. GUIA DE ARQUITETURA E DESIGN DE ÍNDICES DO SQL SERVER. Acessado em 6 de fevereiro de 2020. Disponível em: <<https://docs.microsoft.com/pt-br/sql/relational-databases/sql-server-index-design-guide?view=sql-server-ver15>>.

REZENDE, R. A HISTÓRIA DOS BANCO DE DADOS. Acessado em 3 de março de 2020. Disponível em: <<https://www.devmedia.com.br/a-historia-dos-banco-de-dados/1678>>.

SANCHES, A. R. DISCIPLINA: FUNDAMENTOS DE ARMAZENAMENTO E MANIPULAÇÃO DE DADOS. Disponível em <<https://www.ime.usp.br/~andrers/aulas/bd2005-1/aula3.html>>. Acessado em 8 de abril de 2020.

MACHADO, D. NORMALIZAÇÃO EM BANCO DE DADOS. Disponível em: <https://medium.com/@diegobmachado/normaliza%C3%A7%C3%A3o-em-banco-de-dados-5647cdf84a12>. Acessado em 8 de abril de 2020.

MARCHI, K. NORMALIZAÇÃO. Disponível em: <<http://kessiamarchi.blogspot.com/2013/10/normalizacao.html>>. Acesso em 13 de maio de 2020.

ALVES, G. O QUE É UM BANCO DE DADOS? Disponível em: <<https://dicasdeprogramacao.com.br/o-que-e-um-banco-de-dados/>>. Acesso em 13 de maio de 2020.

# Relatório de Participação dos alunos no estágio