



**ANTONIO MENEGHETTI FACULDADE - AMF**  
**CURSO DE SISTEMAS DE INFORMAÇÃO**

**NATHAMI ROATH RODRIGUES**

**UM ESTUDO SOBRE A OTIMIZAÇÃO E MELHOR PERFORMANCE**  
**DE CONSULTAS NO BANCO DE DADOS SQL SERVER**

**RESTINGA SECA/RS**

**2019**

**NATHAMI ROATH RODRIGUES**

**UM ESTUDO SOBRE A OTIMIZAÇÃO E MELHOR PERFORMANCE  
DE CONSULTAS NO BANCO DE DADOS SQL SERVER**

Trabalho de Conclusão de Curso, apresentado como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação, Curso de Graduação em Sistemas de Informação, Faculdade Antonio Meneghetti - AMF.

Orientador: Profº. Ms. Fabio Sarturi Prass

**RESTINGA SECA/RS**

**2019**

## **AGRADECIMENTOS**

Quero agradecer a Deus por ter me ajudado a superar as dificuldades e a tornar possível a realização de um sonho muito importante para mim.

A esta Instituição eu deixo meu agradecimento, pois sempre encontrei os recursos necessários para evoluir e alcançar todas as metas.

Aos professores e orientador eu deixo uma palavra de agradecimento pela oportunidade, paciência e confiança que depositaram em mim.

Agradeço a minha família, que sempre me incentivou e garantiu que eu não desistisse nunca.

Aos meus amigos deixo aqui minha gratidão, pois foram eles que fizeram com que eu seguisse sempre de cabeça erguida.

Por último quero lembrar aqueles que fizeram parte indiretamente da minha caminhada, e que me incentivaram a correr atrás dos meus sonhos.

Agradeço muito a cada um de vocês!

FACULDADE ANTONIO MENEGHETTI

Nathami Roath Rodrigues

UM ESTUDO SOBRE A OTIMIZAÇÃO E DE CONSULTAS - MELHOR  
PERFORMANCE NO BANCO DE DADOS SQL SERVER

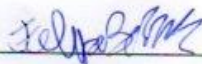
Trabalho de Conclusão de Curso-Monografia, apresentado como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação, Curso de Graduação em Sistemas de Informação, Faculdade Antonio Meneghetti-AMF.

Orientador: Prof. Ms. Fábio Sarturi Prass



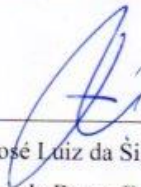
Prof. Ms. Fábio Sarturi Prass

Orientador do Trabalho de Conclusão de Curso  
Antonio Meneghetti Faculdade



Prof. Dr. Felipe Becker Nunes

Membro da Banca Examinadora  
Antonio Meneghetti Faculdade



Prof. José Luiz da Silva Rodrigues Filho

Membro da Banca Examinadora  
Antonio Meneghetti Faculdade

Restinga Seca, RS, 10 de dezembro de 2019

*“O sucesso é a soma de pequenos esforços  
repetidos dia após dia”.*

Roberto Collier.

## RESUMO

O aperfeiçoamento do desempenho de consultas SQL (Structured Query Language) é de extrema importância para os estudantes e profissionais da área de TI (Tecnologia da Informação) e até mesmo de áreas estratégicas dentro de uma empresa que tenham interesse em atuar na área de banco de dados. Isto ocorre, devido à necessidade de obter dados cada vez mais rápidos, de modo a aumentar a produtividade dentro de grandes empresas, estabelecerem laços de fidelidade com os clientes e evitar custos desnecessários com hardware e/ou software dentro de uma empresa, na tentativa de alcançar melhores tempos de respostas através de consultas otimizadas. O projeto deste trabalho de conclusão de curso está focado no principal fator que favorece a aumento do desempenho de consultas SQL. A maneira como são construídas as consultas para atingirem a tempo de resposta desejada, satisfazendo a demanda de solicitações feitas por parte dos usuários junto ao banco de dados.

**Palavras-chave:** Otimização, SQL (Structured Query Language), produtividade, consultas.

## **ABSTRACT**

Improving the performance of Structured Query Language (SQL) queries is of utmost importance to students and IT professionals, and even those within strategic areas within an enterprise who are interested in database operations. This is due to the need for faster and faster data to increase productivity within large enterprises, establish customer loyalty, and avoid unnecessary hardware and / or software costs within an enterprise in an attempt to achieve better response times through optimized queries. The project of this course conclusion work is focused on the main factor that favors the increased performance of SQL queries. The way queries are built to achieve the desired response time, meeting the demand for requests made by users from the database.

**Keywords:** Optimization, Structured Query Language (SQL), query, productivity

## LISTA DE ILUSTRAÇÕES

Figura 1: Banco de Dados .....	19
Figura 2: Os tipos de modelagem do Banco de dados .....	26
Figura 3: Estrutura B-Tree .....	28
Figura 4: Índice em uma Estrutura B-Tree .....	29
Figura 5: Passos do Tuning .....	30
Figura 6: Modelo ER do Banco de Dados utilizado para testes .....	32
Figura 7: Plano de execução - Consulta 1 .....	33
Figura 8: Informação de 'Busca de índices clusterizados' .....	34
Figura 9: Criação de View .....	36
Figura 10: Criação de Índice .....	37
Figura 11: Plano de execução criação do índice .....	37
Figura 12: Resultado da consulta 1 com otimização .....	38
Figura 13: Plano de execução - Consulta 2.....	38
Figura 14: Criação de Índice - Consulta 2.....	39
Figura 15: Plano de execução - Criação do Índice .....	39
Figura 16: Plano de execução - Criação do Índice (cont) .....	39
Figura 17: Plano de execução - Consulta 2 com Índice.....	40
Figura 18: Criação de Índice - Consulta 2.....	40
Figura 19: Plano de execução - Criação do Índice .....	40
Figura 20: Resultado da consulta 2 com otimização .....	41
Figura 21: Plano de execução consulta 3.....	42
Figura 22: Novo rastreamento .....	42
Figura 23: Login SQL Server Profile.....	43
Figura 24: Propriedades do rastreamento .....	43
Figura 25: Informações da consulta .....	44
Figura 26: Salvar rastreamento.....	44
Figura 27: Orientador de Otimização.....	45
Figura 28: Informação do otimizador.....	45
Figura 29: Resultado otimizador.....	46
Figura 30: Criação de índice .....	46
Figura 31: Criação de estatísticas .....	47
Figura 32: Criação de estatísticas 2.....	47
Figura 33: Criação de estatísticas 3.....	47
Figura 34: Resultado consulta 3 com otimização.....	48



## LISTA DE ILUSTRAÇÕES

SQL - *Structured Query Language*

TI – *Tecnologia da Informação*

SGDB – *Data Base Management System*

Modelo ER- *Modelo de entidade relacional*

CPU - *Unidade Central de Processamento*

PK - *Primary Key*

RFID – *Radio Frequency Identification*

# SUMÁRIO

RESUMO	6
ABSTRACT	7
LISTA DE ILUSTRAÇÕES	8
LISTA DE ILUSTRAÇÕES	9
1. INTRODUÇÃO	12
1.1. Objetivos	13
1.1.1. Objetivo Geral	13
1.1.2. Objetivo Específico	13
2. JUSTIFICATIVA	14
3. REFERENCIAL TEÓRICO	16
3.1. Abordagens em Banco de Dados	18
3.2. Tipos Banco de Dados	19
3.2.1. Banco de dados Relacionais e não Relacionais	19
3.2.2. Os principais Bancos de dados	21
3.2.2.1. Oracle	21
3.2.2.2. MySQL	21
3.2.2.3. PostgreSQL	22
3.2.2.4. NoSQL	22
3.2.2.5. MongoDB	22
3.2.3. SQL Server	23
4. METODOLOGIA	25
5. MODELAGEM DE DADOS	26
5.1. Modelo conceitual	26
5.2. Modelo Lógico	27
5.3. Modelo Físico	27
5.4. Técnicas de otimização de instruções SQL	28

6. OTIMIZANDO CONSULTAS SQL COM ALGUMAS FERRAMENTAS NO SQL SERVER	32
7. RESULTADOS OBTIDOS	49
8. CONSIDERAÇÕES FINAIS	51
9. TRABALHOS FUTUROS	52
10. REFERÊNCIAS BIBLIOGRÁFICAS	53

# 1. INTRODUÇÃO

Hoje grandes organizações contam com varias ferramentas que possibilitam o acesso cada vez mais ágil de grandes volumes de dados, informações em tempo hábil. Estas informações não são somente dados, e sim, imagens, vídeos, gráficos, entre outros.

Para Couto (2006), a otimização e melhor desempenho de banco de dados consiste em trazer as informações e resultados para os usuários no menor tempo possível, encontrando assim um caminho mais eficiente para o processamento das mesmas.

Já Chenet al (2006) explica que, o aumento da utilização da tecnologia da Informação (TI) tem resultado na necessidade de avaliar o seu impacto em diversas dimensões como produtividade e qualidade da informação, e para Cohan (2005), esse aumento da utilização acarreta em um grande investimento em TI, justificado pela necessidade de se fornecer informações corretas e precisas ás organizações.

Como cita Gestão (2008), hoje há no mercado de tecnologia diversos sistemas gerenciadores de banco de dados, os quais, na maioria dos casos, são instalados, configurados e utilizados com todos os seus parâmetros em valores padrões, sem levar em consideração o tipo de aplicação para o qual será utilizado, o hardware, e até mesmo o sistema operacional. Com isso, o melhor desempenho do sistema de banco de dados, pode não ser obtido, visto que diversos parâmetros podem ser considerados e ajustados.

Através do estudo do desempenho e otimização de consultas SQL, procura-se a redução do tempo de resposta de um servidor de banco de dados. Essa redução irá influenciar na produtividade de trabalho coletivo à medida que não degrada o desempenho operacional dos sistemas.

Hoje tudo são dados, e as estatísticas e estratégias das empresas são criadas através de uma integração de informações. A grande maioria destes dados chega de forma crua, precisando de tratamento e análise, para que a partir deles já tratados, se transformem em números confiáveis para os executivos das empresas.

O assunto tratado neste trabalho, será a otimização e melhor performance nos resultados de consultas SQL. Serão realizados os testes em consultas rápidas e de grande eficiência para o usuário. Consultas que tragam um grande volume de dados, em um determinado tempo de resultado. Consultas complexas, com ligações entre várias tabelas, que trazem um resultado prático e aplicável.

Serão aplicados comandos SQL, funções e ferramentas para a elaboração das já referidas consultas.

## **1.1. Objetivos**

### 1.1.1. Objetivo Geral

O trabalho tem como objetivo elaborar consultas em SQL, que aperfeiçoam o resultado. Consultas que envolvam mais de uma tabela de um Banco de Dados e através do conhecimento adquirido e estudado, será aplicada em consultas utilizando ferramentas do próprio SQL para aperfeiçoar e melhorar a performance para o resultado.

### 1.1.2. Objetivo Específico

- Compreender os fundamentos de Banco de Dados;
- Apresentar a linguagem SQL utilizada para a construção de consultas;
- Programar consultas SQL que possam servir para demonstrar a otimização das mesmas;
- Descrever as técnicas de otimização empregadas para melhorar o desempenho das consultas;
- Demonstrar, através de ferramentas específicas do software gerenciador de banco de dados utilizado, a execução e monitoramento das consultas;

## 2. JUSTIFICATIVA

O presente trabalho se justifica pelo grande crescimento de dados e informações.

Nos dias atuais a demanda dos Sistemas de Gerenciamento de Banco de Dados está cada vez maior e com isso cresce o volume de dados e a complexidade do software (CARNEIRO; MOREIRA; FREITAS, 2009). O principal desafio tem sido manter o desempenho e eficácia das aplicações, em Sistemas de Gerenciamento de Banco de Dados com grandes volumes de dados, essa que é medida por meio das consultas e alterações (CARNEIRO; MOREIRA; FREITAS, 2009).

Com a melhor performance e otimização de resultados obtidos das consultas SQL criadas, as informações e os dados, serão trazidos de uma maneira mais rápida e eficiente ao usuário. Com as consultas otimizadas será possível, ter essas informações em um curto período de tempo, e através de uma consulta simplificada e de fácil entendimento, trazer milhares de informações em questão de milésimos de segundos ao usuário.

Com a otimização das consultas, as mesmas poderão ser elaboradas de uma forma mais simples, com poucos comandos, pois a otimização serão inclusa dentro das tabelas do Banco de Dados.

Este trabalho utiliza a otimização e melhor desempenho dos resultados, baseados em Bancos de Dados, relativamente grandes, e com muitas informações. Como exemplo de grandes massas de dados, temos o Big Data, que é um termo na tecnologia da informação que se trata de grandes conjuntos de dados que precisam ser tratados, processados e armazenados. Estes dados do Big Data podem ser tanto estruturais quanto não estruturados, que todos os dias são alimentados nas empresas.

O Big Data é uma evolução natural da computação em nuvem, em que o poder de arquivamento e processamento das máquinas migrou para a internet. Os custos são rateados com milhares de usuários (FEIJÓ, 2013).

Falando um pouco sobre o Big Data que possui a definição de '3 v', que são:

- Volume: As organizações coletam os dados de lugares, fontes variadas, incluindo transações financeiras, mídias sociais e informações de sensores ou

dados transmitidos de máquina para máquina. Anteriormente, armazenar essas informações seria um grande problema, mas com as novas tecnologias esse problema quase se sanou.

- **Velocidade:** Os dados são disseminados em uma velocidade sem precedentes e devem ser tratados em tempo hábil para que os usuários não percebam nenhum tipo de *delay*. Etiquetas RFID, sensores e medições inteligentes estão impulsionando a necessidade de lidar com torrentes de dados praticamente em tempo real.
- **Variedade:** As informações são geradas em inúmeros formatos desde estruturados a não-estruturados.

Hoje a maior importância do Big Data é determinar a causa de falhas, erros e irregularidades quase que em tempo real, como gerar um documento fiscal em um ponto de venda com base nos hábitos de compra do cliente, recalcular carteiras de riscos completas em minutos e detectar comportamentos fraudulentos antes que eles afetem sua organização.

Com toda essa grande massa de dados, precisa-se de uma otimização e melhor desempenho nos resultados das consultas que geram esses dados, para que os mesmos sejam mostrados de forma ágil e com sua total integridade ao usuário. Os bancos de dados não relacionais são utilizados para armazenar os dados Big Data, pois não são apenas dados, são textos, imagens, e estas precisam ser mostradas instantaneamente ao usuário, de uma forma que ao inserir uma foto sua em uma rede social, o sistema possa captar os pontos necessários, e o usuário sem perceber aqueles milésimos de segundos, a foto será a associada à pessoa correta, trazendo todas as informações cadastradas.

### 3. REFERENCIAL TEÓRICO

Para (Gonçalves, 2006), a otimização de consultas SQL tem o objetivo produzir a escolha mais eficiente para execução dos comandos SQL, o que determina em um ótimo desempenho para a aplicação.

Segundo Date (2004) a otimização representa um desafio para os sistemas de banco de dados relacionais, se o sistema espera atingir um desempenho aceitável. A vantagem não está apenas no fato que os usuários não precisam se preocupar em formular consultas para tirar o melhor desempenho do sistema, mas na possibilidade do otimizador sair-se melhor que o usuário.

De acordo com Carneiro et al (s.d):

O primeiro caminho para conseguir um desempenho adequado de um sistema de banco de dados é tomar boas decisões durante o projeto desse. Várias considerações deverão ser feitas durante a fase de projeto, entre elas: o volume esperado de dados em cada relação do sistema e quais consultas serão realizadas com mais frequência. Mas, percebe-se na maioria dos sistemas é que seu real desempenho só pode ser conseguido após algum tempo de uso, e muitas das considerações que os projetistas haviam feito podem mostrar-se incorretas. Portanto, uma fase subsequente de ajuste do sistema torna-se necessária, com base em dados reais de seu comportamento, com o objetivo de maximizar o desempenho e a estabilidade.

Elmasri e Navathe (2011), eles citam que um banco de dados é uma grande quantidade de dados relacionados, ou seja, o termo “dado” significa fatos conhecidos que podem ser marcados, e possuem significado implícito e podem ser interpretados.

Já na visão de Vinhas (2016) um banco de dados pode ser definido como uma série de dados organizados de uma forma que um computador possa guardá-los e recuperá-los de maneira eficiente e rápida. O autor ainda explica que bancos de dados transformaram os computadores em repositórios de informações.



Para Date (2003) que afirma que um banco de dados é uma coleção de dados persistentes, usados por sistemas de aplicação de uma determinada empresa.

Ou seja, o mais importante é ter um bom monitoramento do sistema de gerenciamento de dados e, compreender o funcionamento de consultas SQL, para que seja possível realizar as consultas do melhor desempenho possível.

### 3.1. Abordagens em Banco de Dados

O Banco de dados é simplesmente um conjunto de dados. Por esse ponto, pode se tratar de uma planilha com uma lista de usuários ou clientes. Precisa-se de uma abordagem numerosa, para ser considerado o Banco de dados.

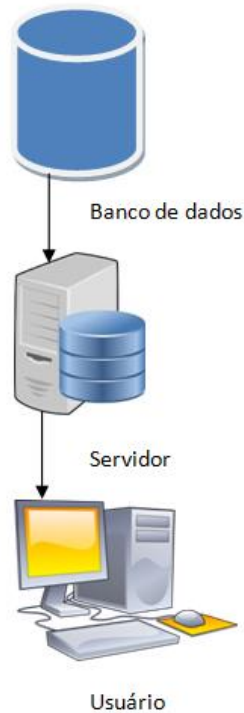
Um banco de dados deve reunir informações de maneira organizada, consistente, íntegra e acessível em tempo hábil. É neste ponto que temos o conceito de Sistema de Gerenciamento de Banco de Dados (SGBD).

Basicamente, um Sistema de Gerenciamento de Banco de Dados é uma estrutura de software que permita que dados sejam armazenados, organizados, mantenham sua integridade, sejam atualizados, acrescentados, excluídos e acessados sempre que tiver houver necessidade, tendo que corresponder à demanda que a aplicação que o utiliza exige.

No banco de dados é possível colocar dados à disposição de usuários para uma consulta, uma introdução, responsabilizando pelos direitos atribuídos aos mesmos. Isso é mais importante ainda quando os dados informados são de grande volume. Um banco de dados pode ser local, ou seja, utilizável em um dispositivo por um usuário, ou repartido, isto é, quando as informações são armazenadas em dispositivos remotos e acessíveis pela rede, como exemplo um Banco de Dados armazenado em nuvem. A sua grande eficácia do uso dos bancos de dados é a possibilidade de ter mais de um usuário com o mesmo acesso, em tempo real.

Na Figura 1 observa-se como é a estrutura de Banco de dados, Servidor e Usuário, de uma forma simples e clara, e ligação entre eles.

Para que seja possível ter o controle dos dados assim como dos usuários, um sistema de gerenciamento deve ser criado. A gestão do banco de dados é feita em um sistema chamado SGBD (sistema de gerenciamento de bancos de dados). O SGBD é um conjunto de serviços que permite gerenciar os bancos de dados, ou seja, dar acesso aos dados de uma forma mais fácil, autorizar o acesso às informações a múltiplos usuários e manipular os dados que estão atualmente no banco de dados como inserção, supressão e modificação.



**Figura 1: Banco de Dados**

## **3.2. Tipos Banco de Dados**

### **3.2.1. Banco de dados Relacionais e não Relacionais**

Com a grande expansão da produção de dados, é também cada vez maior a responsabilidade dos especialistas em Big Data. Para que seja possível executar as suas funções com aplicabilidade, os profissionais precisam conhecer as diferenças entre banco de dados relacional e não relacional.

Bancos de dados relacionais são fundamentados no paradigma da orientação a conjuntos. Seus dados são armazenados em estruturas com nomes de tabelas e cada tabela é composta por colunas, tuplas ou registros. Também no modelo relacional, primeiramente toda a estrutura do banco de dados deve ser projetada, porque se essas tabelas não forem

definidas, não é possível inserir os dados nas mesmas. Alguns exemplos de banco de dados relacionais:

- Oracle
- MySQL
- PostgreSQL
- SQLite
- SQL Server

Os Bancos de dados não relacionais, e o NoSQL, podem ser visto como um padrão de armazenado alternativo ao modelo SQL (relacional), podendo oferecer uma maior escalabilidade. Esses modelos não relacionais surgiram exatamente pela necessidade de se ter uma melhor performance e alta escalabilidade, para dessa forma ter um maior e melhor gerenciamento de dados.

Utilizando NoSQL, não se tem a necessidade de fazer todo o esquema antes de começar a utilizá-lo, até porque todas as informações serão agrupadas em um registro, então não é necessário que se tenha relacionamentos entre várias tabelas para ser formada a informação, ela estará em sua totalidade no mesmo registro.

Como no MongoDB, que ele utiliza Json. Alguns exemplos de banco de dados não relacionais:

- MongoDB
- Cassandra
- Voldemort
- CouchDB
- Riak

### 3.2.2. Os principais Bancos de dados

#### 3.2.2.1. Oracle

Oracle Database é o SGDB mais utilizado no mundo. Foi lançado no final dos anos 70, tendo como linguagem de programação oficial o PL/SQL. Suas funcionalidades priorizam a integridade dos dados e tem disponível uma ampla gama de recursos. Trata-se de um Banco robusto, confiável e seguro, podendo ser instalados em múltiplas plataformas, como Unix, Linux, HP/UX, BIM AIX, IBM VMS e Windows. Porém, é necessário que seja investido em hardware para não prejudicar o desempenho da aplicação. Possui documentação bastante detalhada, o que permite que o usuário ou desenvolvedor tenha todo conhecimento necessário dos recursos deste banco.

A Oracle conta com recursos de segurança e desempenho considerados essenciais para empresas que têm aplicações críticas e muitos dados. Por esse motivo é um dos mais indicados para grandes empresas ou aplicações que possuem requisitos de negócios mais complexos.

#### 3.2.2.2. MySQL

Também pertencente à Oracle. Um banco de dados relacional Open Source (código aberto), cujo seu objetivo são os sistemas online. Utiliza a linguagem SQL (Structured Query Language – Linguagem de Consulta Estruturada) como interface. Funciona sob as licenças de software livre e comercial. Com comprovado desempenho, confiabilidade e facilidade de uso, tornou-se a primeira opção para aplicativos baseados na Web, é utilizado pelos principais sites, como o Facebook, Twitter, YouTube, Google e NASA.

### 3.2.2.3. PostgreSQL

Um banco de dados relacional Open Source, desenvolvido pela PostgreSQL Global DevelopmentGroup. Por ser Open Source, é também outra opção muito utilizada para sistemas Web, e por grandes empresas como Apple, Skype e o Metrô-SP. É um dos sistemas de gerenciamento de dados mais avançados, com recursos como consultas complexas, chaves estrangeiras, facilidade de acesso e integridade transacional. Tem muita semelhança com o Oracle, devido à sua linguagem e estrutura. As principais vantagens em seu uso estão relacionadas à economia e ao alto desempenho oferecido pelo SGBD.

### 3.2.2.4. NoSQL

O termo NoSQL foi criado em 1998 como um banco de dados não relacional de código aberto. A partir da popularização das redes sociais e do avanço da geração de conteúdo por dispositivos, o armazenamento de dados com o objetivo de utilizá-los em ferramentas analíticas passou a acarretar em maiores custos. A opção por utilizar bancos de dados não relacionais favorece uma escalabilidade mais fácil e econômica, pois não exige um maquinário muito poderoso. Além disso, sua facilidade de manutenção proporciona uma redução na equipe. Por isso, os bancos de dados NoSQL se tornaram populares entre as grandes empresas.

### 3.2.2.5. MongoDB

MongoDB é um dos bancos de dados NoSQL mais utilizados, open source e se encontra disponível para Windows, Linux e OSX. Seu lançamento ocorreu em fevereiro de 2009 pela empresa 10gen, e sua linguagem de programação é o C++, o que garante ótima performance. É orientado a documentos (documentdatabase) no formato JSON. Isso significa que não apresenta como restrição a necessidade de ter tabelas e colunas criadas

previamente, o que permite que um documento represente toda a informação necessária no formato de um JSON. MongoDB foi criada com Big Data em mente, e suporta escalonamento horizontal ou vertical.

### 3.2.3. SQL Server

O SQL Server é um banco de dados relacional muito utilizado no mercado, criado pela Microsoft em 1988. Sua linguagem de programação é o T-SQL. Oferece recursos avançados que facilitam a atualização dos dados e garantem a confiabilidade das informações armazenadas. Trata-se de um Banco de Dados muito seguro que atua com sistemas integrados de criptografia, o que garante que os dados somente serão visualizados ou alterados por usuários autorizados. É possível criar regras de integridade para que não autorize que sejam excluídos dados importantes, como por exemplo, um departamento que tenha dados bancários de clientes. Esse fato se dá para que não exista no banco de dados uma informação sem a sua relação correspondente.

No domínio corporativo, o SQL Server é usado por empresas de vários portes e segmentos, com destaque para indústrias, bancos e instituições governamentais. Além disso, é um dos mais usados em sites de e-commerce e tendo como competidores os sistemas como o MySQL e Oracle.

O SQL Server é um SGBD (sistemas de gerenciamento de Banco de Dados) da Microsoft, criado em parceria com a Sybase, em 1988, inicialmente como um complementar do Windows NT, sendo que depois passou a ser aperfeiçoado e vendido separadamente. A parceria com a Sybase terminou em 1994, e a Microsoft continuou a melhorar o programa após isto.

Esse SGBD é dos mais usados no mundo atualmente, O SQL Server tem versões gratuitas e pagas. O preço da versão paga do SQL Server é bem menor que a média do mercado, mesmo com excelente qualidade. O SQL é bastante usado em sites, onde são necessários cadastros, e também em sistemas de lojas, onde são lançados os produtos, o preço, marca entre outras informações.

O SQL Server é um gerenciador de bancos de dados relacional, o que significa dizer que as informações que manipula estão armazenadas em campos de tabelas. Para facilitar o entendimento, imagine, por exemplo, uma tabela de clientes, em que cada linha contém os registros de nome, endereço, cidade, estado e país. São estruturas como essa que o banco controla. Adicionalmente, o sistema oferece recursos avançados para facilitar a atualização dos dados e garantir que as informações armazenadas se mantenham corretas e confiáveis. Entre as funcionalidades pode-se destacar:

- **Segurança:** O SQL Server atua com sistemas de criptografia integrada, garantindo que os dados somente serão visualizados ou alterados por usuários que possuem autorização expressa para tal. É possível que o administrador do sistema defina permissões para acesso apenas a determinadas tabelas.
- **Integridade:** Com o uso de controles sobre os dados, o SQL Server impede que sejam geradas inconsistências que inviabilizam a utilização precisa das informações. Tem-se um exemplo prático: imagine que exista uma tabela de funcionários, em que um dos campos seja o código do departamento em que trabalha. Outra tabela, de departamentos, contém o código e nome do respectivo departamento. Em função das regras de integridade nativas do SGBD, um usuário não conseguirá excluir um departamento que possua funcionários associados. Isso é feito de forma transparente, para que não exista uma informação sem a correspondente relação no banco de dados, ou seja, um código de departamento sem seu nome na tabela devida.
- **Concorrência:** O banco de dados da Microsoft possibilita que vários usuários acessem a mesma tabela ao mesmo tempo. Porém existe um controle inteligente para manter a consistência das informações, isto é, para garantir que os dados estejam devidamente atualizados. Isso é obtido com o uso de bloqueios parciais, alertas e também da gravação de versões anteriores dos dados, como forma de consolidar a base de dados.



## 4. METODOLOGIA

No presente trabalho será utilizada como metodologia, a explicativa, que tem como objetivo descrever os fatores que determinam ou são de relevância para a ocorrência dos fenômenos. Tem um amplo aprofundamento do conhecimento da realidade porque explica a razão e o porquê dos eventos. Quando esse tipo de pesquisa é realizado no campo da tecnologia, são utilizados métodos de pesquisa e observação.

Na primeira etapa deste projeto serão reunidas as fontes bibliográficas e materiais disponibilizados em sites da internet. Na segunda etapa, através da leitura e estudo das diversas fontes bibliográficas obtidas no item acima, será abordado o embasamento teórico do projeto. Serão abordados os fundamentos de banco de dados, a linguagem SQL, as técnicas de otimização e o sistema gerenciador de banco de dados.

Na terceira etapa deste projeto, será criado o banco de dados SQL Server com dados e informações contábeis. Este banco de dados será criado em uma máquina local, onde sua única função será a execução do banco de dados.

Após a criação do Banco de Dados SQL Server com os dados de uma empresa X de Contabilidade serão criadas as consultas. No início as consultas serão básicas, para que seja possível estimar o tempo de entrega dos dados, a partir desse primeiro tempo, será realizada as otimizações nas consultas para que o tempo de entrega seja cada vez menor.

Cada consulta criada terá um resumo, explicando as técnicas e ferramentas utilizadas, mostrando o tempo de execução de cada consulta através do plano de execução, e com as informações destes resultados será possível analisar e mostrar qual é a técnica mais eficiente para o cliente/empresa.

## 5. MODELAGEM DE DADOS

Modelagem de dados é o ato de explorar estruturas orientadas a dados. Como outros artefatos de modelagem, modelos de dados podem ser usados para uma variedade de propósitos, desde modelos conceituais de alto nível até modelos físicos de dados. Com a modelagem de dados é possível identificar os tipos de entidades da mesma forma que na modelagem de classes identificamos classes. Atributos de dados são associados a tipos de entidades exatamente como associados atributos e operações às classes. Existem associações entre entidades, similar às associações entre classes, relacionamento, herança, composição e agregação são todos conceitos aplicáveis em modelagem de dados, conforme Figura 2.

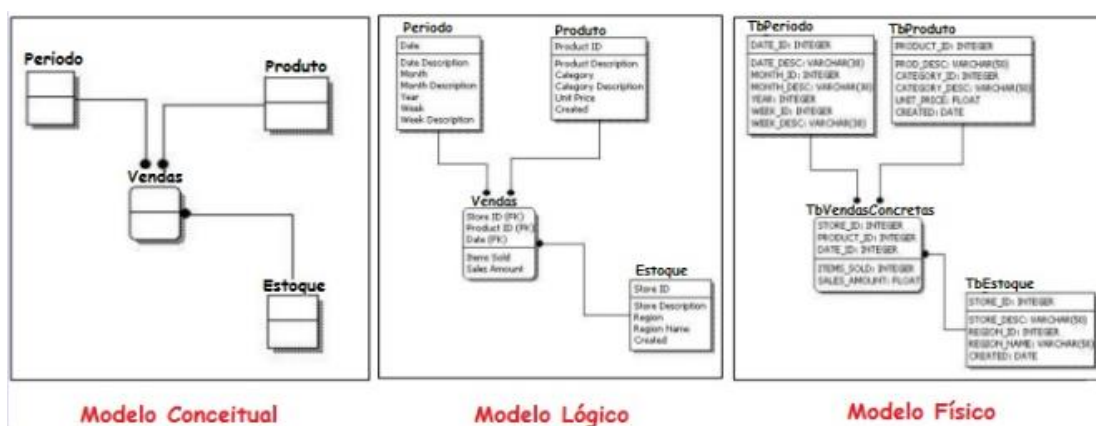


Figura 2: Os tipos de modelagem do Banco de dados

### 5.1. Modelo conceitual

Este modelo representa as regras de negócio sem limitações tecnológicas ou de implementação por isto é a etapa mais adequada para o envolvimento do usuário que não precisa ter conhecimentos técnicos.

- Visão Geral do negócio
- Facilitação do entendimento entre usuários e desenvolvedores
- Possui somente as entidades e atributos principais

- Pode conter relacionamentos n para m.

## **5.2. Modelo Lógico**

Leva em conta limites imposto por algum tipo de tecnologia de banco de dados.

- Deriva do modelo conceitual e via a representação do negócio
- Possui entidades associativas em lugar de relacionamentos n:m
- Define as chaves primárias das entidades
- Normalização até a terceira forma normal
- Adequação ao padrão de nomenclatura
- Entidades e atributos documentados

## **5.3. Modelo Físico**

Leva em consideração limites imposto pelo sistema de gerenciamento de banco de dados e pelos requisitos não funcionais dos programas que acessam os dados.

Características:

- Elaborado a partir do modelo lógico
- Pode variar segundo o SGBD
- Pode ter tabelas físicas (log , líder , etc.)
- Pode ter colunas físicas (replicação)

## 5.4. Técnicas de otimização de instruções SQL

Segundo Pletsch (2005), a escolha correta de quais colunas irá formar o índice é algo fundamental para a sua aplicabilidade. Cabe ao projetista saber identificar se é necessária a criação de novos índices e quais serão os critérios adotados para a sua criação.

A ferramenta adotada para melhorar o desempenho das consultas será o Índice, que são utilizados para facilitar a busca de informações em uma tabela com o menor número possível de operações de leituras, tornado assim a busca mais rápida e eficiente. O SQL Server utiliza o mesmo princípio da lista telefônica gravando as informações dos índices em uma estrutura chamada de B-Tree conforme Figura 3.

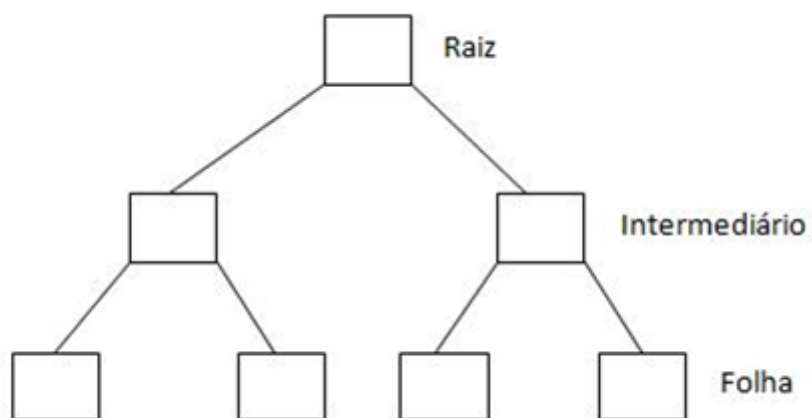
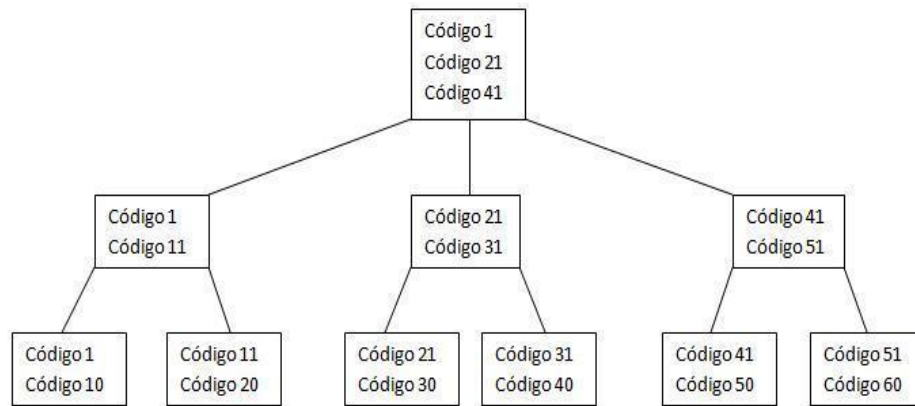


Figura 3: Estrutura B-Tree

Na Figura 4 mostra-se um exemplo de índice em uma estrutura B-Tree para um campo código do tipo inteiro.



**Figura 4: Índice em uma Estrutura B-Tree**

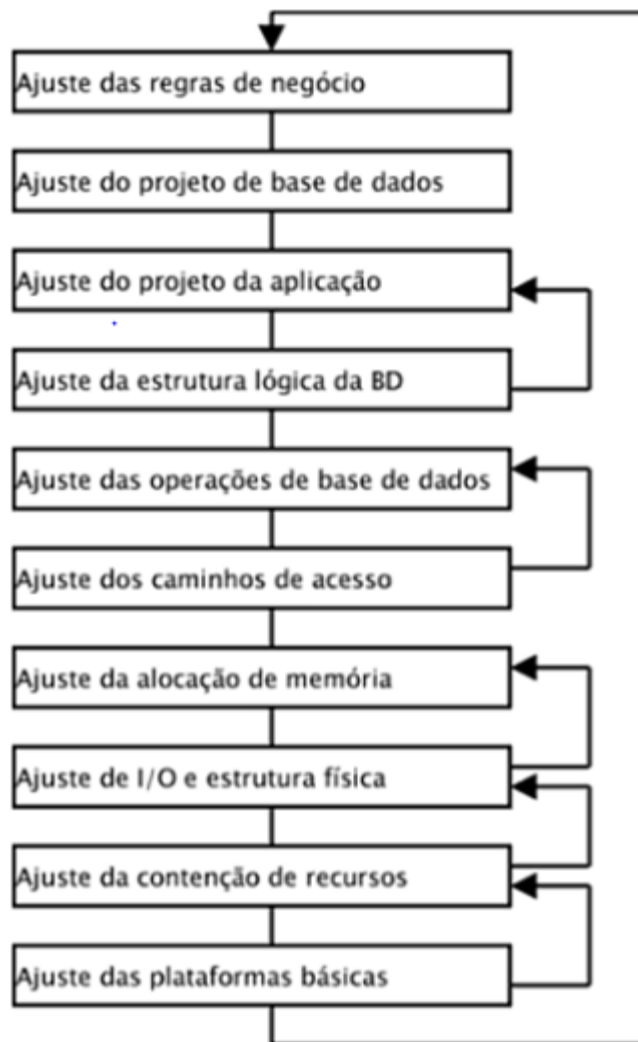
Na Figura 4, para encontrar o código 23 iniciaria a busca pelo nível raiz percorrendo as linhas. Como o código 23 está entre 21 e 41 o SQL Server calcula que o código 23 se encontra na seqüência do código 21 e pula para a página do nível intermediário que contém os valores 21 a 31. Em seguida analisaria que a primeira opção (21) se encaixa para a busca e pularia para a página de nível folha que contém a cadeia de 21 a 30, percorreria a mesma até achar o código 23 e finalizaria a busca. No SQL Server é possível criar índices clusterizados (*clustered*), não clusterizados (*nonclustered*), XML e espaciais, sendo os dois últimos para melhorar os recursos de pesquisas em documentos XML e aplicações espaciais, e neste artigo os mesmos não serão abordados.

A técnica de otimização com Views é uma representação virtual de uma tabela, e pode conter colunas de uma ou varias tabelas físicas ou até mesmo de outras Views. Na maioria das vezes, as Views não armazenam os dados em banco, elas consultam os dados que foram selecionados em suas tabelas de origem. É possível utilizar uma View em qualquer lugar onde se possa utilizar uma tabela.

As técnicas de Tuning também são utilizadas para otimizar e melhorar os resultados das consultas SQL. Com esta ferramenta é possível executar o rastreador de otimização ao mesmo tempo em que a consulta está sendo executado no SQL Server, após a finalização da mesma, será possível acessar o software SQL Server Profile e analisar o desempenho da

consulta, e as informações para criação de índices e estatísticas para melhorar e deixar o resultado da consulta de uma forma mais otimizada.

Na Figura 5, são mostrados os passos do Tuning.



Fonte Ikematu 2009

Figura 5: Passos do Tuning

O tuning como observado na Figura 5 pode ser realizado em diferentes SGBDR, a maioria destes, possui parâmetros semelhantes que podem ser modificados para alcançar o desempenho pretendido. Alguns manuais de SGBD trazem a descrição de parâmetros que podem ser alterados para melhorar seu desempenho, o tempo de execução das consultas e transações do SGBDR (ELMASRI; NAVATHE,2004).

Tuning significa sintonia ou ajuste para que algo funcione melhor. Ele oferece um suporte ao Administrador do Banco de Dados (DBA), fazendo ajustes nas aplicações que melhoram a performance do banco de dados (Ikematu 2009).

O tuning de SGBD procura resolver diretamente problemas como, a concorrência entre as transações, reduzir a sobrecarga de registro de logs que aumentam o armazenamentos desnecessários de dados, otimizar o escalonamento de processos, e o tamanho de buffer, e por fim destinar recursos como disco, memória e processos buscando a eficiência na sua aplicação. (CARNEIRO; MOREIRA; FREITAS, 2007).

## 6. OTIMIZANDO CONSULTAS SQL COM ALGUMAS FERRAMENTAS NO SQL SERVER

Neste tópico serão mostradas as informações do Banco de Dados SQL Server, criado para as execuções de testes de otimizações das consultas.

Foi criado o Banco de Dados, com informações de Contabilidade de uma empresa X. A principal tabela utilizada nas consultas, será a tabela 'Lançamento', está que possui 33904568 registros. Por ser uma tabela que possui muitos registros e possui ligação com diversas tabelas distintas, será possível criar as consultas e otimizá-las. A tabela 'Lançamento', possui os dados mais importantes deste banco de dados, com os principais dados do usuário, como endereço, telefone, números de notas fiscais, comprovantes e pagamentos com suas respectivas datas.

Na Figura 6, é mostrado o Modelo ER do Banco de Dados, com as tabelas utilizadas nas consultas SQL.

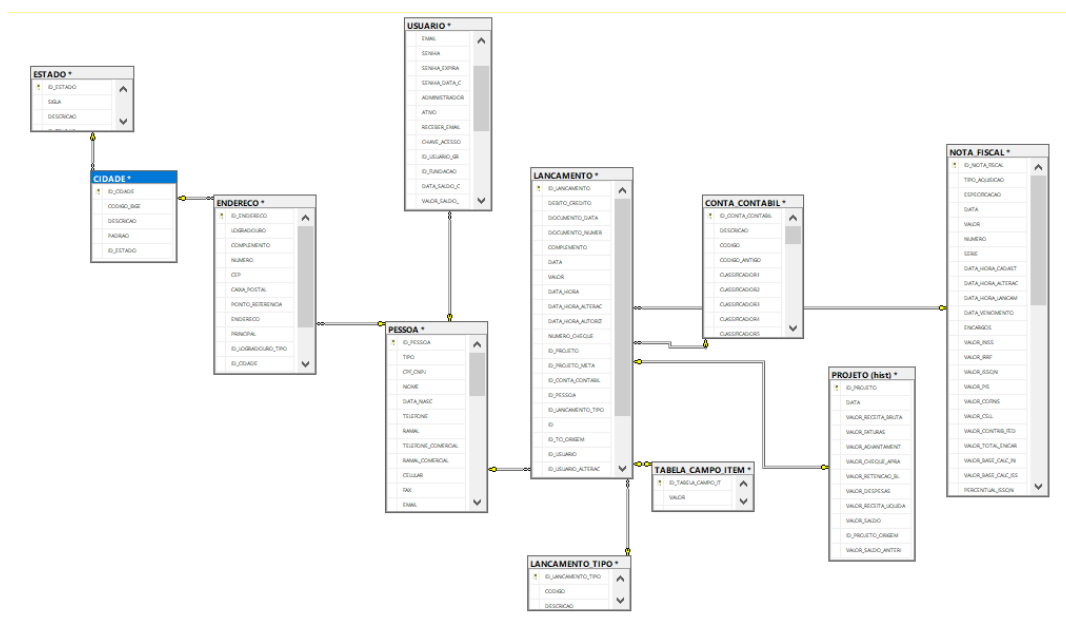


Figura 6: Modelo ER do Banco de Dados utilizado para testes

Serão criadas as consultas SQL e executadas, para que seja possível mostrar o tempo de resultado de uma consulta simples em comparação a uma consulta otimizada.



Para se analisar o real ganho de performance usou-se a ferramenta *query optimizer*, a qual contém os planos baseado em custo são gerados, o que significa que entre várias possibilidades uma é escolhida baseada no menor custo. Os cálculos vão levar em consideração algumas informações como, por exemplo, recursos de CPU, I\O e estatística. As estatísticas por padrão são coletadas automaticamente, baseadas nas colunas dos índices dentro do banco de dados elas descrevem a distribuição dos dados, a unicidade, ou seletividade destes.

O caminho ótimo para execução da consulta será definido pelo *optimizer* tendo como resultado o plano de execução. Neste processo serão definidos quais os tipos de *joins* serão usados, quais índices serão acessados e ainda outras informações que são baseadas em custo.

Na Figura 7 observa-se o plano de execução de consulta sem a aplicação de otimização. O tempo de execução foi de 00h01min25, para trazer um total de 71 linhas, um tempo consideravelmente alto, para a quantidade de dados trazidos. Conforme mostrado no plano de execução é mostrado dois Índices, que são as chaves primárias das tabelas (PK).

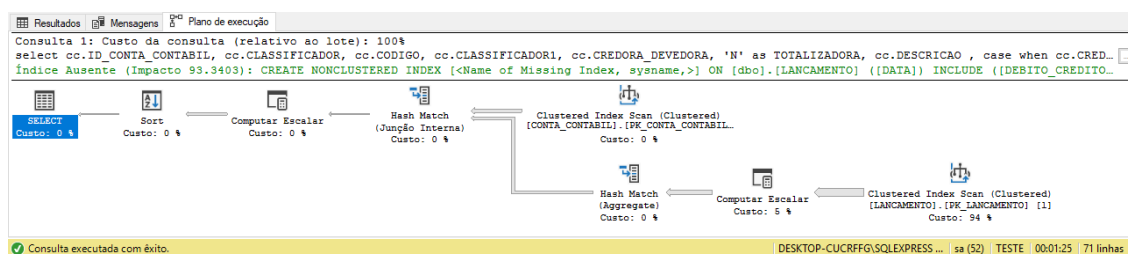


Figura 7: Plano de execução - Consulta 1

Ao passar o mouse em cima do item 'Busca de índices clusterizados PK\_LANCAMENTO- 52%' são possíveis analisar as informações de consumo para tal consulta.

Na Figura 8 de cada item tem uma porcentagem que representa o custo relativo da consulta, para cada operador representado pelos ícones. Este custo vai de 0% até 100%, uma consulta pode ter múltiplos passos, cada passo vai ter um valor de custo relativo dentro do contexto da consulta.

Clustered Index Scan (Clustered)	
Exame de um índice clusterizado, completo ou somente de um intervalo.	
Operação Física	Clustered Index Scan
Operação Lógica	Clustered Index Scan
Modo de Execução Real	Row
Modo de Execução Estimado	Row
Armazenamento	RowStore
Número de Linhas Lidas	33904568
Número Real de Linhas	1365168
Número Real de Lotes	0
Custo Estimado de E/S	479,733
Custo Estimado do Operador	517,028 (52%)
Custo Estimado de CPU	37,2952
Custo Estimado da Subárvore	517,028
Número Estimado de Execuções	1
Número de Execuções	1
Número Estimado de Linhas	1418860
Número Estimado de Linhas a serem Lidas	33904600
Tamanho Estimado da Linha	208 B
Reassociações Reais	0
Retrocessos Reais	0
Ordenado	True
ID do Nó	3
<b>Predicado</b>	
datepart(year,[TESTE].[dbo].[LANCAMENTO].[DATA] as [].[DATA])= (2019)	
<b>Objeto</b>	
[TESTE].[dbo].[LANCAMENTO].[PK_LANCAMENTO] [I]	
<b>Lista de Saída</b>	
[TESTE].[dbo].[LANCAMENTO].ID_LANCAMENTO; [TESTE].[dbo].[LANCAMENTO].DEBITO_CREDITO; [TESTE].[dbo].[LANCAMENTO].DOCUMENTO_DATA; [TESTE].[dbo].[LANCAMENTO].DOCUMENTO_NUMERO; [TESTE].[dbo].[LANCAMENTO].COMPLEMENTO; [TESTE].[dbo].[LANCAMENTO].DATA; [TESTE].[dbo].[LANCAMENTO].VALOR; [TESTE].[dbo].[LANCAMENTO].DATA_HORA; [TESTE].[dbo].[LANCAMENTO].DATA_HORA_ALTERACAO; [TESTE].[dbo].[LANCAMENTO].DATA_HORA_AUTORIZACAO; [TESTE].[dbo].[LANCAMENTO].NUMERO_CHEQUE; [TESTE].[dbo].[LANCAMENTO].ID_PROJETO; ...	

Figura 8: Informação de 'Busca de índices clusterizados'

Ao passar com o mouse pelos ícones percebe-se uma janela, como pode ser visualizada na Figura 8, esta janela é a *tooltip*. Cada operador tem uma *tooltip* associada a ele, com detalhes referente à execução da *package*. A natureza de execução de cada operador define um grupo de dados que estará disponível nesta *tooltip*.

As seguintes informações podem aparecer nas *tooltips*:

- **CachedPlanSize** (Operador *Select*): Tamanho do plano armazenado no *Cache*;
- **LogicalOperation**: Representa o resultado dos cálculos do *Optimizer* para o que deve acontecer quando consulta executar;
- **PhysicalOperation**: Representa o que realmente aconteceu quando a *query* foi executada. Geralmente *Physical* e *Logical* são iguais;
- **ActualNumberRows** (disponível se a opção *IncludeActualExecutionPlan* for selecionada): Número atual das linhas retornadas baseada em quantas vezes a função *GetNext()*, o qual retorna uma linha por dado, foi chamada;

- Estimated I/O Cost: Baseado em cálculos feitos pelo *Optimizer*, é apenas uma estimativa, mas pode ser um ponto de atenção para consultas com alto custo de I/O;
- Estimated CPU Cost: Baseado em cálculos feitos pelo *Optimizer*, é apenas uma estimativa, mas pode ser um ponto de atenção para consultas com alto custo de CPU;
- EstimatedOperatorCost: Estimativa de custo para a execução do operador. Ex: Estimativa de custo para um *tableScan*, ou seja, para a leitura de toda a tabela linha por linha sem uso de índice;
- EstimatedSubtreeCost: Custo acumulado da execução dos passos anteriores até o passo atual (leitura deve ser feita da direita para a esquerda);
- EstimatedNumberofRows: Cálculo baseado na estatística disponível para *Optimizer* produzindo a estimativa de número de linhas. Esta varia de ícone a ícone os quais representam os operadores envolvidos no plano de execução. Cada operador está lidando com um grupo e quantidade diferente de dados;
- EstimatedRowSize: Estimativa do tamanho da linha retornada, baseado em estatística;
- ActualRewinds (disponível se a opção *Include ActualExecutionPlan* for selecionada): Significa que um ou mais parâmetros correlacionados ao *join* mudaram e o *inner* precisa ser reavaliado. Refere-se ao número de vezes que o método *INIT()* foi chamado;
- ActualRewinds (disponível se a opção *Include ActualExecutionPlan* for selecionada): Significa que um ou mais parâmetros correlacionados ao *join* mudaram antes do resultado do *inner* e este precisa ser usado novamente. Refere-se ao número de vezes que o método *INIT()* foi chamado;
- Ordered: Valor do tipo *boolean*, o qual informa se os dados estão ordenados ou não. Em alguns casos as informações retornadas por uma consulta vão determinar que os dados ou um grupo deles estivessem organizados de maneira ordenada, isto pode gerar um processamento extra no plano;

- Node ID: Simplesmente indica a ordem do nó, ou seja, a ordem dos operadores. Perceba que eles estão ordenados da esquerda para a direita, embora a leitura do plano seja melhor da direita para a esquerda.

Ao clicar com o botão direito no plano de execução no modo gráfico e selecionar propriedades, pode-se obter mais detalhes sobre as operações. A maioria do que está disponível nas propriedades já é conhecido, exceto por algumas informações, como por exemplo:

- Forced Index: Valor *boolean*, que vai ficar *true* se a consulta estiver usando algum *hint* para forçar o uso de índice;
- No ExpandHint: Mesmo conceito do *ForcedIndex* aplicado a *indexedviews*;
- Parallel: valor lógico do tipo *boolean*, podendo assumir os valores 0 e 1. 0 indica que não está rodando em paralelo, e 1 que está rodando em paralelo.

Através da consulta mostrada na Figura 9, como primeira opção de otimização, foi criado uma *View*. Nota-se que o tempo de resposta da consulta aumentou para 00h01min26seg, ou seja, para esta consulta a criação de uma *View* não foi válida. Conforme mostrado no Plano de execução da consulta, são mostradas as informações de ‘Índice Ausente’, e o seu impacto de 93.3837% na consulta.

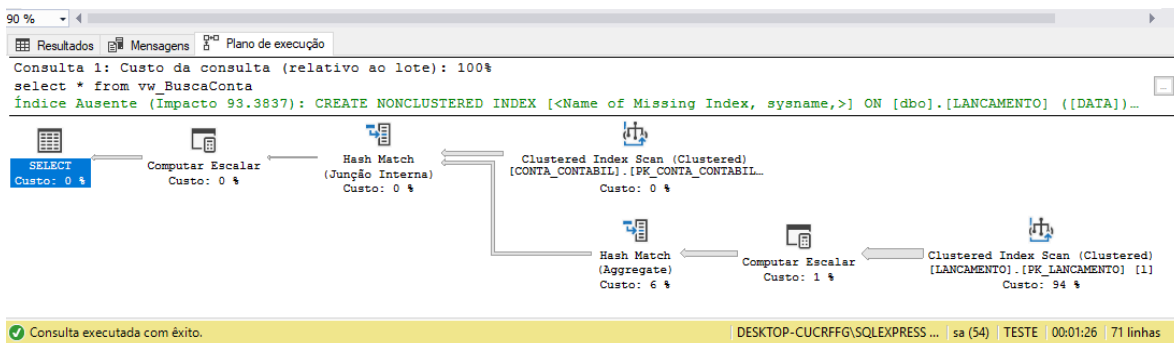


Figura 9: Criação de View

O Processador de Consultas estima que a implementação do índice a seguir possa melhorar o custo da consulta em 86.1273%.

Com a informação mostrada no plano de execução foi criado o índice, conforme descrito na Figura 10.

```
17 | create nonclustered index idxLancamentoConta on dbo.lancamento([data] include ([debito_credito])
```

Figura 10: Criação de Índice

Na Figura 11 é mostrado o plano de execução da criação do índice da Figura 10.

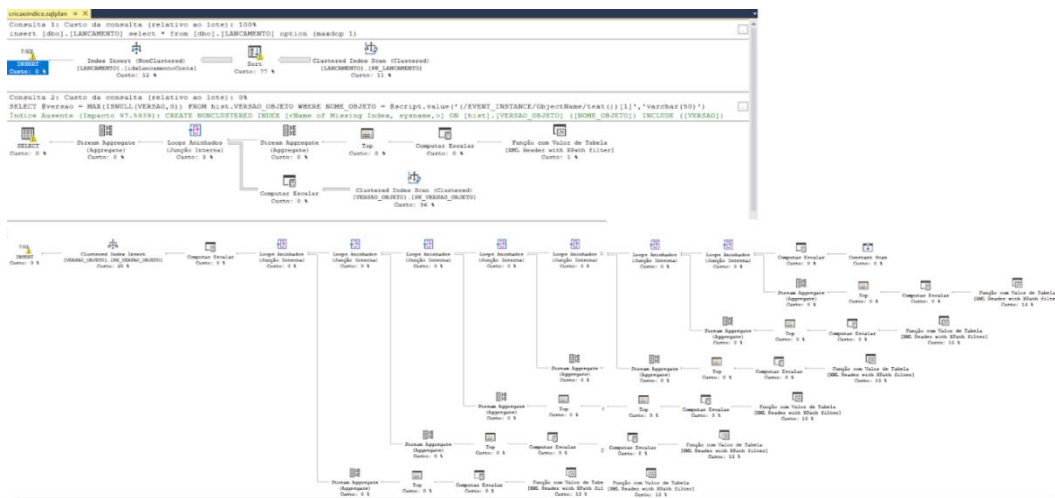


Figura 11: Plano de execução criação do índice

Com a aplicação de índices, a consulta levou 00h00min03seg em sua execução, conforme plano de execução da Figura 12.

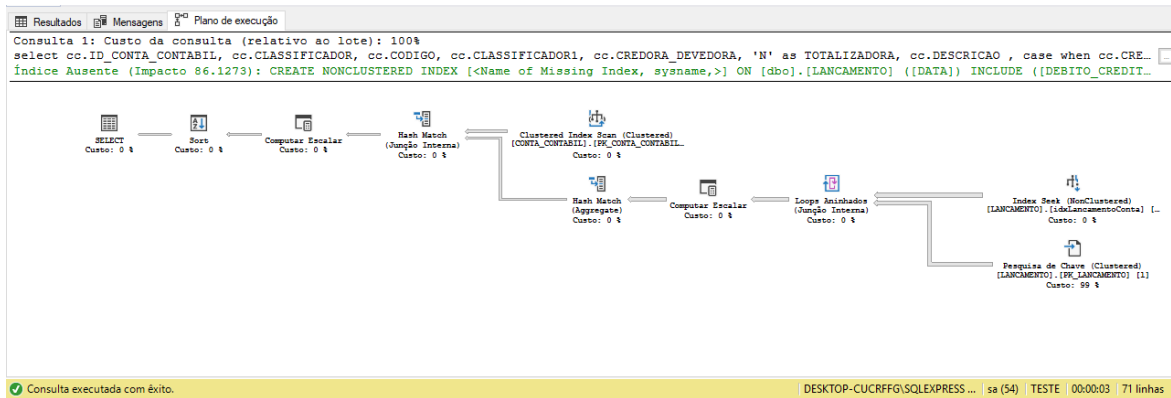


Figura 12: Resultado da consulta 1 com otimização

Ao passar o mouse em cima da informação *Index Seek* (Non Clustered) que mostra as linhas e páginas que se qualificam com essas linhas (resultado). O seu custo é de 0%, proporcional de acordo com a quantidade de linhas e páginas qualificadas ao invés de ler todas as linhas da tabela. Índices *Seeks* geralmente são preferidos por sua alta capacidade de seleção nas consultas.

Na Figura 13, foi criada uma consulta simples, sem otimizações. O objetivo desta consulta é trazer os clientes com moram no endereço com o CEP '97010423', ordenado pelo nome, data da nota fiscal de lançamento e número da nota fiscal. Nota-se que esta consulta trouxe 9832 linhas em 00h01min34seg. No plano de execução é possível analisar as tabelas que necessitam de aplicação da otimização para melhorar o resultado da consulta.

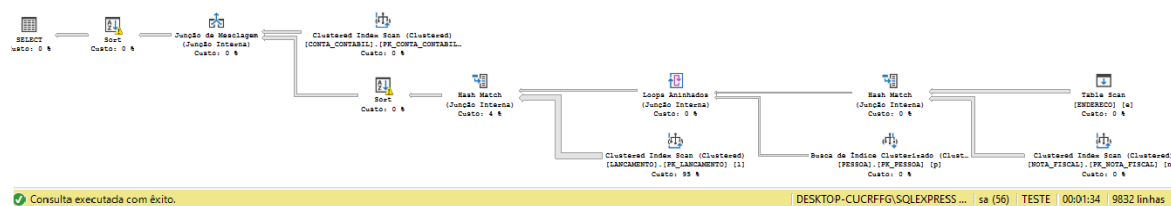


Figura 13: Plano de execução - Consulta 2

Para otimizar a consulta mostrada na Figura 13, foi criado um índice para na tabela ENDERECO, em coluna CEP. Na Figura 14 é mostrado o script da criação do índice.

```

CREATE NONCLUSTERED INDEX IDX_ENDERECO_CEP ON ENDERECO
(
    CEP ASC
)
WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)

```

Figura 14: Criação de Índice - Consulta 2

Na Figura 15 é mostrado o plano de execução da criação do índice.

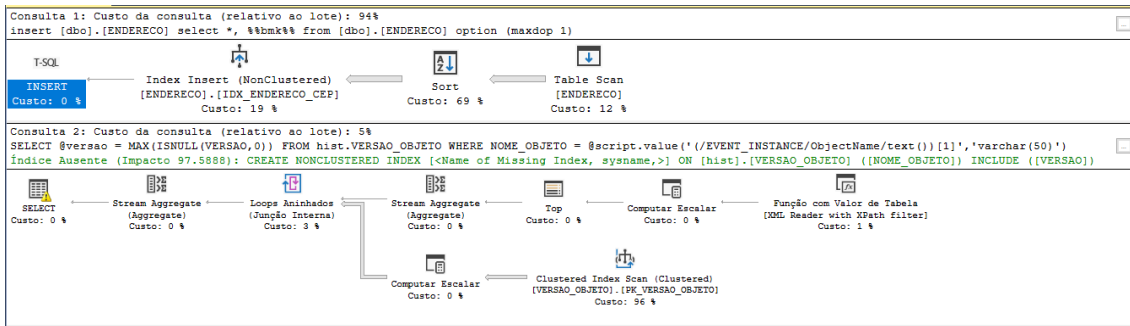


Figura 15: Plano de execução - Criação do Índice

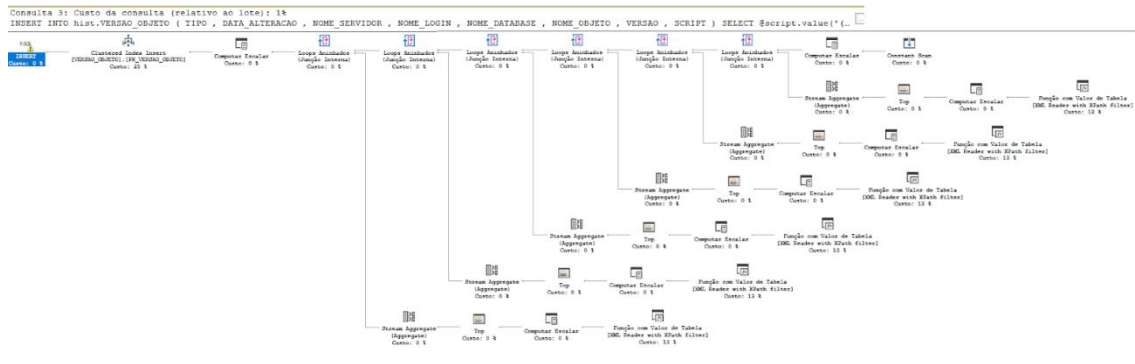


Figura 16: Plano de execução - Criação do Índice (cont)

Na Figura 16 segue a continuação do plano de execução da criação do índice.

Após a criação do índice, a consulta foi novamente executada Figura 17, e tempo de resultado diminuiu 18 segundos. Novo tempo de resultado 00h01min16seg.

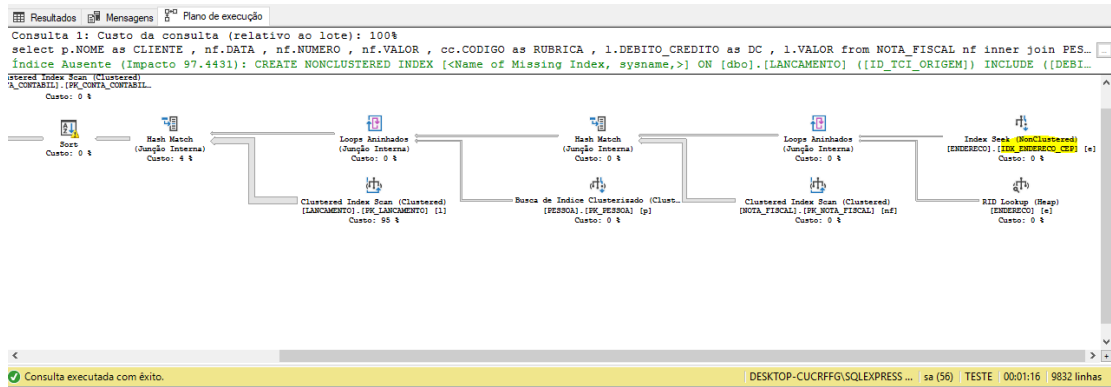


Figura 17: Plano de execução - Consulta 2 com Índice

Foi criado um novo índice na tabela ‘LANCAMENTO’, para melhorar ainda o desempenho da consulta, conforme Figura 18.

```
CREATE NONCLUSTERED INDEX IDX_LANCAMENTO_ID ON LANCAMENTO
(
    ID, ID_TCI_ORIGEM
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
```

Figura 18: Criação de Índice - Consulta 2

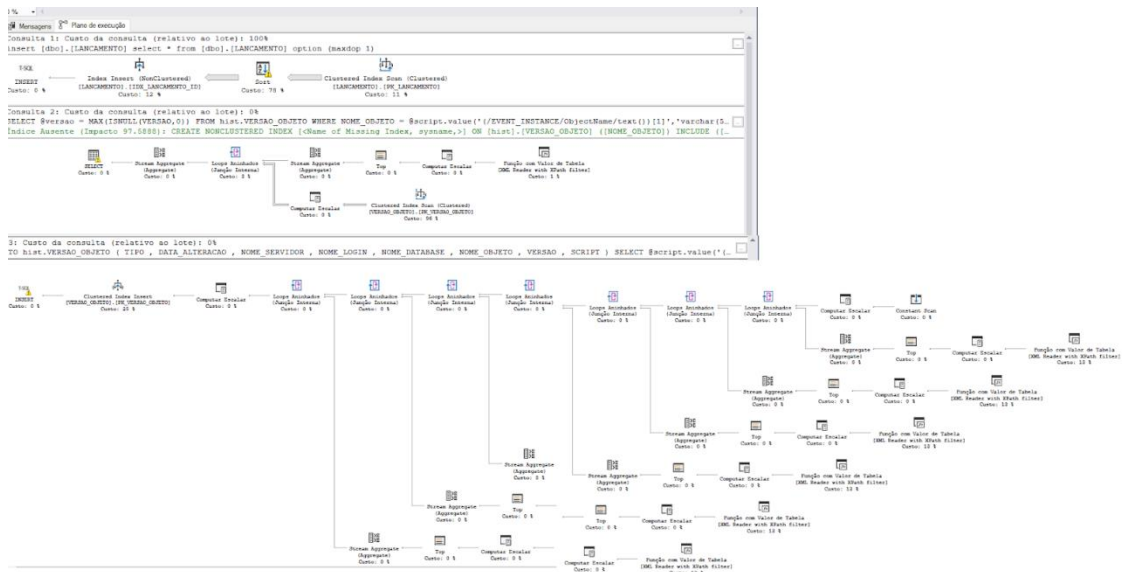


Figura 19: Plano de execução - Criação do Índice



Na Figura 19 é mostrado o plano de execução da criação do índice, referente à Figura 18.

Após a criação deste novo índice, foi executada novamente a consulta. Conforme mostrado na Figura 20 nota-se que a consulta trouxe o mesmo número de linhas, em um tempo de 00h00min09seg. Com essa otimização, a consulta teve um ganho de tempo de 00h07min66seg.

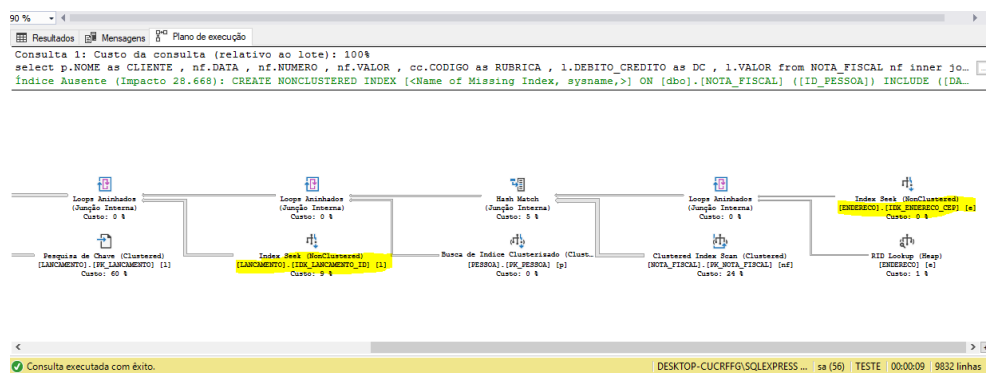


Figura 20: Resultado da consulta 2 com otimização

Para a utilização da técnica de Tuning, foi necessária a instalação do software SQL Server Profiler, que é uma interface para criar e gerenciar rastreamentos, além de analisar e reproduzir resultados de rastreamento. Os eventos serão salvos em um arquivo de rastreamento que posteriormente pode ser analisado ou utilizado para reproduzir uma série específica de etapas na tentativa de diagnosticar um problema. Para que seja possível verificar quais colunas devem ser otimizados, e também consumo que aquela consulta está tendo no Banco de dados.

Para essa análise de Tuning, foi criada consulta conforme Figura 21, com tempo de resposta de 00h02min17seg e para 1365168 linhas. Através do plano de execução da consulta, temos em 'Loops Aninhados' com o custo de 1% e 3% em junções internas que seriam os comandos de *Inner Joins* utilizado na consulta.

Para os índices já existentes as porcentagens variam, em 'Busca de Índice Clusterizado [USUARIO].[PK\_USUARIO] [U] tem o custo de 21%. Clustered index Scan

[LANCAMENTO].[PK\_LANCAMENTO] custo de: 54%. Busca de Índice clusterizado [LANCAMENTO\_TIPO].[PK\_LANCAMENTO\_TIPO] custo de 21%.

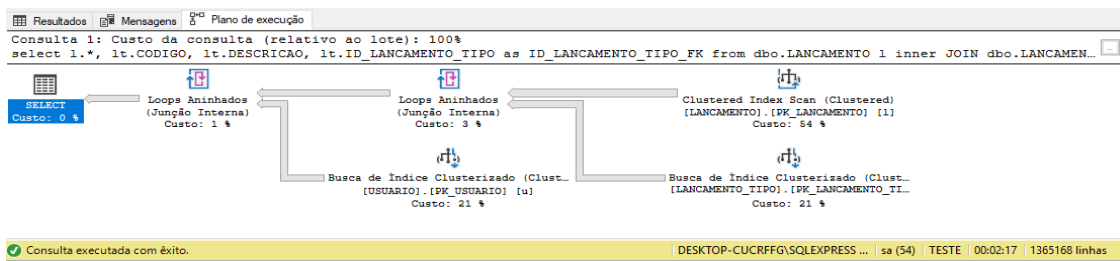


Figura 21: Plano de execução consulta 3

Para analisar novamente o tempo de execução da consulta da Figura 21, e onde deverá ser otimizada, será acessado software já instalado na máquina 'O SQL Server Profiler'.

Conforme mostrado na Figura 22, será necessário abrir o software SQL Server Profiler, clicar em Arquivo -> Novo rastreamento.

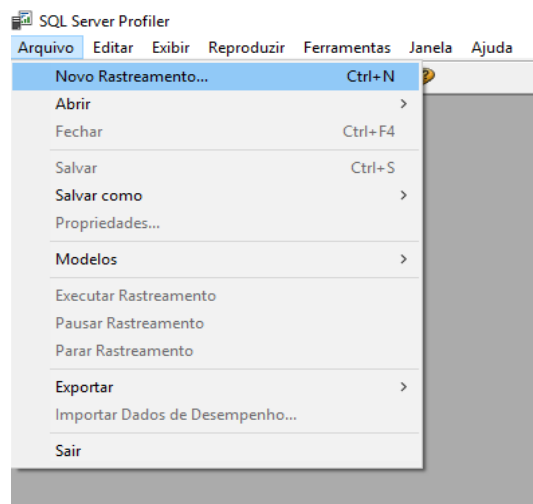


Figura 22: Novo rastreamento

Realizar o login conforme Figura 23. Os dados de login serão os mesmos para o login do SQL Server.

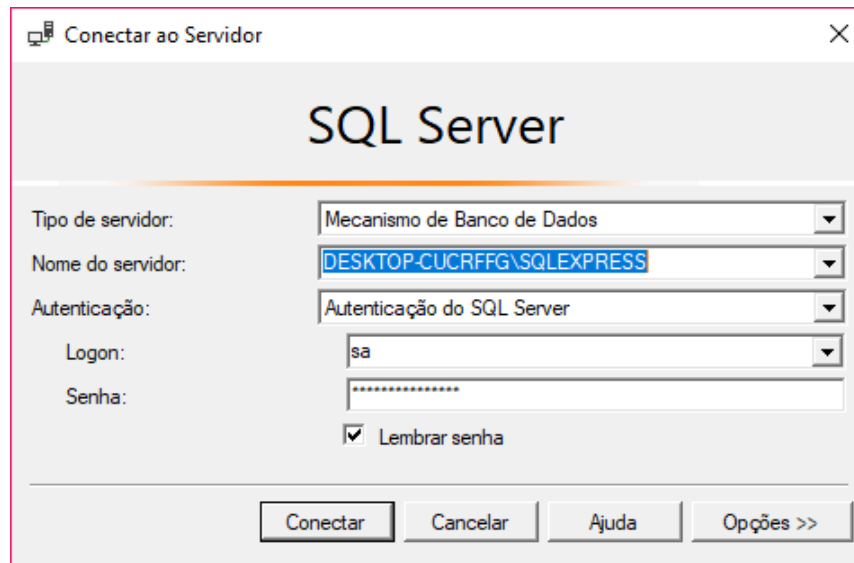


Figura 23: Login SQL Server Profile

Na Figura 24 é mostrada a página de ‘Propriedades do Rastreamento’, na opção de ‘Usar o modelo’, escolher a opção ‘Tuning’, e clicar em executar. A partir deste momento o SQL Server Profile vai começar a rastrear o Banco de dados.

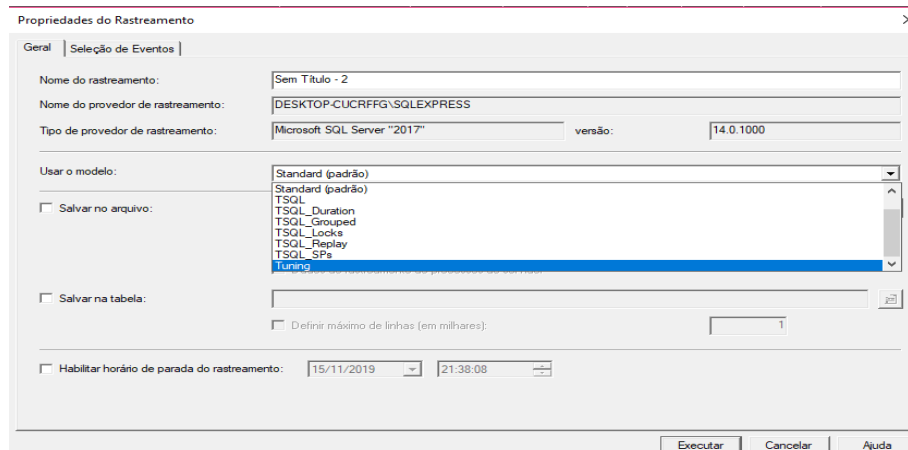


Figura 24: Propriedades do rastreamento

Após a execução, deve-se acessar novamente a consulta no SQL Server Management Studio e executar - lá. As informações mostrada no SQL Server Profile, são de tempo de execução da consulta em milissegundos na coluna ‘Duration’, na coluna

‘TextData’ é mostrado as informações de linhas e também duração da mesma, conforme Figura 25.

EventClass	TextData	Duration	SPID	DatabaseID	DatabaseName	Object Type	LoginName
Trace Start							
RPC:Completed	exec sp_trace_setstatus 4,1	0	56	1	master		sa
SQL:BatchCompleted	SELECT @SPID;	0	54	19	TESTE		sa
SQL:BatchCompleted	SET STATISTICS XML ON	0	54	19	TESTE		sa
SQL:BatchCompleted	-- 00:02:27.1365168 11linhas select 1...	121762	54	19	TESTE		sa
SQL:BatchCompleted	SET STATISTICS XML OFF	0	54	19	TESTE		sa

Figura 25: Informações da consulta

Deve-se salvar o arquivo de rastreamento conforme Figura 26, para que seja possível testar o mesmo, e analisar em qual tabela e coluna deverá ser realizada a otimização para melhorar o desempenho da consulta.

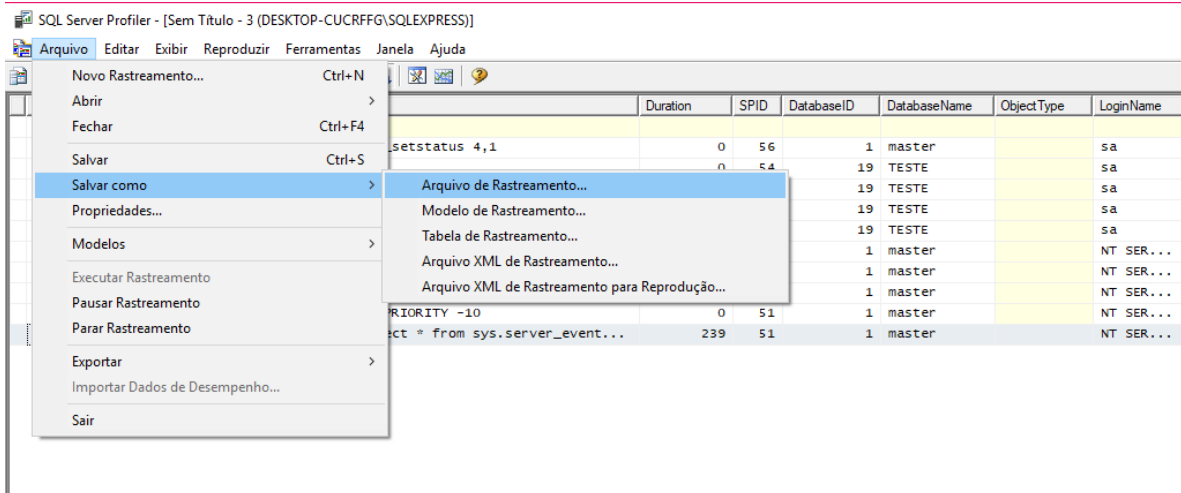


Figura 26: Salvar rastreamento

Após salvar o arquivo de rastreamento conforme Figura 27, é necessário acessar no menu ‘Ferramentas -> ‘Orientador de Otimização do Mecanismo de Banco de Dados’, e localizar o arquivo salvo na maquina local.

EventClass	Text	Duration (s)	DatabaseID	DatabaseName	ObjectType	LogName
Trace:Start						
RPC:Completed		6	1	master		sa
SQL:BatchCompleted		4	19	TESTE		sa
SQL:BatchCompleted	SET STATISTICS XML ON	0	19	TESTE		sa
SQL:BatchCompleted	-- 00:02:17 1365168 linhas select 1...	121752	19	TESTE		sa
SQL:BatchCompleted	SET STATISTICS XML OFF	0	19	TESTE		sa
SQL:BatchCompleted	SET DEADLOCK_PRIORITY -10	0	51	1	master	NT SER...
SQL:BatchCompleted	SELECT target_data FROM sys.dm...	114	51	1	master	NT SER...
RPC:Completed	exec sp_reset_connection	0	51	1	master	NT SER...
SQL:BatchCompleted	SET DEADLOCK_PRIORITY -10	0	51	1	master	NT SER...
SQL:BatchCompleted	if exists(select * from sys.server_event...	239	51	1	master	NT SER...
RPC:Completed	exec sp_reset_connection	0	51	1	master	NT SER...
SQL:BatchCompleted	SET DEADLOCK_PRIORITY -10	0	51	1	master	NT SER...
SQL:BatchCompleted	SELECT target_data FROM sys.dm...	129	51	1	master	NT SER...
RPC:Completed	exec sp_reset_connection	0	51	1	master	NT SER...
SQL:BatchCompleted	SET DEADLOCK_PRIORITY -10	0	51	1	master	NT SER...
SQL:BatchCompleted	if exists(select * from sys.server_event...	260	51	1	master	NT SER...

Figura 27: Orientador de Otimização

Na Figura 28, são mostradas as informações de estatísticas e índices para criação, para que seja possível otimizar e melhorar o desempenho do resultado da consulta.

Database Name	Object Name	Recommendation	Target of Recommendation	Details	Partition Scheme	Size (KB)	Definition
TESTE	[dbo].[LANCAMENTO]	create	_lta_index_LANCAMENTO_14_594413487_K16_K19_K6_K1			753504	(ID_LANCAMENTO_TIPO) asc, (ID_USUARIO) asc, (DATA) asc, (ID_LANCAMENTO) asc
TESTE	[dbo].[LANCAMENTO]	create	_lta_stat_594413487_1_5				(ID_LANCAMENTO) (DATA)
TESTE	[dbo].[LANCAMENTO]	create	_lta_stat_594413487_19_6				(ID_USUARIO) (DATA)
TESTE	[dbo].[LANCAMENTO]	create	_lta_stat_594413487_6_16_19_1				(DATA, ID_LANCAMENTO_TIPO, ID_USUARIO, ID_LANCAMENTO)
TESTE	[dbo].[USUARIO]	create	_lta_index_USUARIO_14_947078710_K1			16	(ID_USUARIO) asc

Figura 28: Informação do otimizador

Na Figura 29 é mostrado a criação de uma estatística (CREATE STATISTICS), O otimizador de consulta usa as estatísticas para criar planos de consulta que melhoram o desempenho das consultas.

eme	Size (KB)	Definition
	753504	[[ID_LANCAMENTO_TIPO] asc, [ID_USUARIO] asc, [DATA] asc, [ID_LANCAMENTO] asc]
		[[ID_LANCAMENTO], [DATA]]
		[[ID_USUARIO], [DATA]]
		[[DATA], [ID_LANCAMENTO_TIPO], [ID_USUARIO], [ID_LANCAMENTO]]
	16	[[ID_USUARIO] asc]

```

CREATE STATISTICS [_dta_stat_594413487_1_6] ON [dbo].[LANCAMENTO]([ID_LANCAMENTO], [DATA])

```

**Figura 29: Resultado otimizador**

Com a consulta sendo executado juntamente com o Orientador de Otimização do Mecanismo de Banco de Dados, foi informado que será necessária a criação de um índice com a tabela USUARIO (ID\_USUARIO), conforme Figura 30.

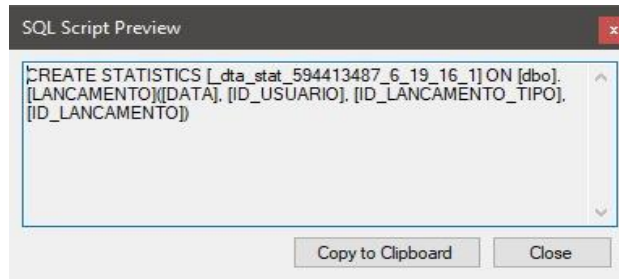
```

CREATE NONCLUSTERED INDEX
[_dta_index_USUARIO_14_947078710__K1] ON [dbo].[USUARIO]
(
    [ID_USUARIO] ASC
)
WITH (SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE
= OFF) ON [PRIMARY]

```

**Figura 30: Criação de índice**

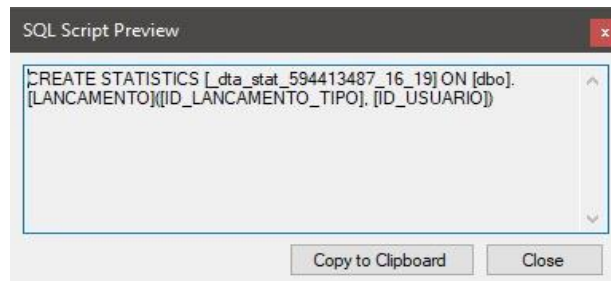
Através do otimizador, também foi sugerido à criação de três estatísticas mostrado nas Figuras 31, 32 e 33 para melhorar o resultado da consulta, que são nas tabelas LANCAMENTO (DATA, ID\_USUARIO, ID\_LANCAMENTO\_TIPO, e ID\_LANCAMENTO).



```
CREATE STATISTICS [_dta_stat_594413487_6_19_16_1] ON [dbo].[LANCAMENTO]([DATA], [ID_USUARIO], [ID_LANCAMENTO_TIPO], [ID_LANCAMENTO])
```

Copy to Clipboard Close

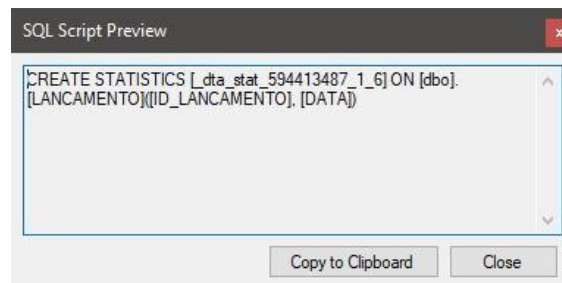
Figura 31: Criação de estatísticas



```
CREATE STATISTICS [_dta_stat_594413487_16_19] ON [dbo].[LANCAMENTO]([ID_LANCAMENTO_TIPO], [ID_USUARIO])
```

Copy to Clipboard Close

Figura 32: Criação de estatísticas 2



```
CREATE STATISTICS [_dta_stat_594413487_1_6] ON [dbo].[LANCAMENTO]([ID_LANCAMENTO], [DATA])
```

Copy to Clipboard Close

Figura 33: Criação de estatísticas 3

Após a criação dos índices e das estatísticas, foi possível notar que o tempo de resultado desta consulta baixou para 00h02min04seg, uma redução importante para o número de dados trazidos. Através do plano de execução da consulta, temos em '*Loops Aninhados*' com o custo de 1% e 3% em junções internas que seriam os *Inner Joins* utilizado na consulta.

Para os índices já existentes as porcentagens variam, em 'Busca de Índice Clusterizado [USUARIO].[PK\_USUARIO] [U] tem o custo de 21%. Clustered index Scan

[LANCAMENTO].[PK\_LANCAMENTO] custo de: 54%. Busca de Índice clusterizado [LANCAMENTO\_TIPO].[PK\_LANCAMENTO\_TIPO] custo de 21%.

Ao analisar o plano de execução da Figura 34 com o primeiro sem a otimização, é notado que os valores das porcentagens em alguns pontos diminuiram como no [USUARIO] que de 21%, o custo baixou para 0%. Porém em [LANCAMENTO\_TIPO] teve um aumento para 23% o seu custo.

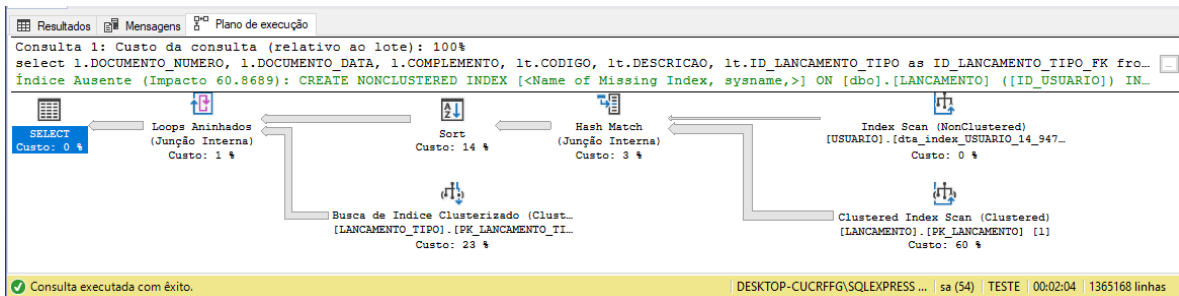


Figura 34: Resultado consulta 3 com otimização



## 7. RESULTADOS OBTIDOS

Com as consultas realizadas, foi possível identificar uma grande melhora no desempenho dos resultados das consultas aplicando as otimizações estudadas.

Em uma análise da primeira consulta sem nenhuma otimização é mostrado o resultado foi de 00h01min25, para trazer um total de 71 linhas, como primeira opção de otimização foi realizada a criação de uma View e executada novamente, porém o resultado não foi como o esperado, pois o tempo de resposta da consulta aumentou em 1 segundo (00h01min26seg). Ao verificar o plano de execução da consulta é mostrada uma sugestão de índice para a consulta, que melhoraria o custo da consulta em 86.1273%. Diante desta informação, foi criado mais o índice na consulta, onde o resultado da resposta da mesma foi de 00h00min03seg, deixando a consulta otimizada e rápida. Nota-se neste primeiro teste, que um ponto negativo, foi à criação de uma View, que seu efeito sob o resultado não foi satisfatório. Com isso, é possível perceber que realizar algumas otimizações na tentativa e erro, no falso positivo, pode atrapalhar os resultados e em consequência, fazer com que o usuário perca tempo em algo que não poderá não dar certo.

Para a segunda consulta, o primeiro resultado sem otimização foi de 9832 linhas em 00h01min34seg. Como alternativa de otimização, foi criado o primeiro índice na tabela ENDERECO, está que está vinculado a todos os clientes, pois está com seus dados de endereço, e a coluna CEP que também é vinculada a todos os clientes e fornecedores, e o resultado foi de uma melhora de desempenho de 18 segundos (00h01min16seg). Novamente foi analisado o plano de execução e realizada a criação de um novo índice LANCAMENTO, e com esse novo índice o resultado do tempo de resposta da consulta com os dados foi de 00h00min09seg. Com essa otimização, a consulta teve um ganho de tempo de 00h07min66seg.

Para a terceira consulta, foi aplicada a técnica de Tuning, para que seja possível através de um software o SQL Server Profile, seja possível analisar quais são os possíveis índices e estatísticas a serem criadas. Notou-se que com Tuning, as informações de desempenho da consulta são mais claras e corretas. A consulta simples, sem nenhuma otimização teve o tempo de resposta de 00h02min17seg e para mostrar 1365168 linhas.

Após criação de índices e estatísticas, a consulta teve seu retorno em 00h02min04seg, uma boa melhora para a quantidade de dados que a consulta entrega.

## 8. CONSIDERAÇÕES FINAIS

Este trabalho de conclusão de curso apresentou um estudo sobre Banco de Dados, tendo como objetivo principal a otimização e melhor desempenhos nos resultados de consultas no Banco de Dados SQL Server, utilizando técnicas e ferramentas como índices, views, técnica de tuning.

Com as execuções das consultas não otimizadas foi analisado o plano de execução de cada uma, e através dele, foi possível identificar onde melhorar o desempenho nos resultados de cada consulta.

Com esta análise realizada a através do plano de execução, foi possível analisar os resultados também através de porcentagens e não somente do tempo, pois como os testes foram realizados em uma maquina local, poderia haver divergências no tempo a cada execução da mesma consulta.

A principal ferramenta utilizada foi à criação de índices, pois apenas com eles, foi possível reduzir o tempo de execução das consultas. Com a criação de um ou mais índices em uma consulta, pode melhorar cada vez mais o desempenho da mesma.

Neste banco de dados, foi constatado que as *views* não tiveram muitos efeitos positivos, pois quando foram aplicadas, em alguns casos tornou o resultado da consulta um pouco mais demorado.

O Tuning realizado no Banco de Dados é de extrema importância, pois através dele podemos melhorar o desempenho do sistema de gerenciamento de dados, configurando o mesmo para utilizar forma mais eficiente os seus recursos e os oferecidos pelo hardware onde este se encontra. A utilização deste método possibilita alcançar aspectos mais específicos do desempenho nas consultas.

## **9. TRABALHOS FUTUROS**

Como possíveis trabalhos futuros, pode-se apontar:

Aprofundamento na técnica de Tuning e ferramentas utilizadas para sua execução, pois com essa ferramenta, a empresa que utilizará poderá ter um pouco de certeza de como otimizar o resultado da sua consulta SQL.

Um estudo mais amplo na criação de índices baseados nos resultados dos planos de execuções de cada consulta, através das porcentagens mostradas.

## 10.REFERÊNCIAS BIBLIOGRÁFICAS

ALECRIM, Emerson. **Bancos de dados são mais importantes nas nossas vidas do que a gente imagina**. 2018. Artigo (Negócios) - [S. l.], 2018. Disponível em: 2018. Acesso em: 26 jun. 2019.

AUGUSTO WEBER, FLAVIO. **ANALISE DE DESEMPENHO E OTIMIZAVAO DE CONSUITAS SQL**. 2002. Monografia (Grau no Curso de Tecnologia em Processamento de Dados da Universidade Tuiuti do Parana) - Curitiba, 2002.

CARNEIRO, Alessandro Pinto; MOREIRA, Julinao Lucas; FREITAS, André Luis Castro de. **TUNING - Técnicas de Otimização de Banco de Dados**, Um Estudo Comparativo: Mysql e Postgresql. Trabalho de Graduação - Centro de Ciências Computacionais – Universidade Federal do Rio Grande (FURG).

FEIJÓ, Bruno Vieira. **A Revolução dos Dados**. *Revista Exame PME – Pequenas e Médias Empresas*, São Paulo, p. 30-43, set. 2013.

CHEN, Y, et al. Evaluationofinformationtechnologyinvestment: a data envelopmentanalysis approach. **Computers&OperationsResearch**, New York, May 2006

COHAN, P. S. CFOsto tech: III spendfrtherightttechnology.**Financial Executive**, Morristown, v.21 n.3,p.30-34, Apr 2005.

ELOISE FLÔR, Daniela. **TÉCNICAS, METODOLOGIAS E FERRAMENTAS PARA PROJETO DE BANCOS DE DADOS DISTRIBUÍDOS: UM ESTUDO DE SOLUÇÕES PROPOSTAS**. 2001. Dissertação (Grau mestre ciência da computação) - Florianópolis, 2001.

**GESTAO, REVISTA CIENTIFICA DE ADMINISTRAÇÃO E SISTEMAS DE INFORMAÇÃO/UNIDADE DE ENSINO SUPERIOR**, Expoente. V.11, n.11. Jul. /Dez. 2008

ÍNDICES no SQL Server. 2010. - Artigo, [S. l.], 2010. Disponível em: <https://www.devmedia.com.br/indices-no-sql-server/18353>. Acesso em: 30 set. 2019.

LOPES, Nicholas. **Índices no SQL Server**. Disponível em: <http://www.devmedia.com.br/indices-no-sql-server/18353>. Acesso em: 31 mai. 2013.

MUXFELDT., Pedro. **Banco de Dados**. 2017. Artigo (CCM (<https://br.ccm.net/>) - [S. l.], 2017. Disponível em: <https://br.ccm.net/contents/65-bancos-de-dados>. Acesso em: 27 jun. 2019.

NOGARE, Diego. **Melhorando desempenho de consultas utilizando Views Indexadas**. Artigo (Ciencia da Computacao)-Disponível em: <http://www.linhadecodigo.com.br/artigo/1308/melhorando-desempenho-de-consultas-utilizando-views-indexadas.aspx>. Acesso em: 30 set. 2019.

PLETSCH, Edson Luis. **Avaliação das Técnicas de Desempenho em Sistemas Gerenciadores de Banco de Dados Relacionais**. Trabalho de Conclusão de Curso - Centro Universitário Feevale Instituto de Ciências Exatas e Tecnológicas Curso de Ciência da Computação, 2005.

PORNI, Ricardo Portilho, Artigo SQL Magazine: Oracle Benchmark: Linux x Windows

VINHAS, Lúvia. Fundamentos de bancos de dados. Earth System Science Centre. 2016. Disponível em: <<http://ess.inpe.br/courses/lib/exe/>>