Diagrama

Descrição gerada automaticamente

Universidade Estadual de Campinas

FACULDADE DE TECNOLOGIA

**PROJETO DE OTIMIZAÇÃO DE BANCO DE DADOS DE CALL CENTER**

**Danilo Matos**

**Gustavo Ferreira Lima**

**Jackson Rodrigues**

**Lucas Ferreira**

**Mateus de Almeida Frigo**  
Curso: Engenharia e Administração de Banco de Dados

Professor(a): Manoel A. da Silva Jr

Campinas/2024

**SUMARIZAÇÃO DOS RECURSOS DE OTIMIZAÇÃO NO MODELO STAR SCHEMA  
  
Atendimentos - Call Center**  
**Modelo Entidade-Relacionamento no Esquema Estrela**

1. **Modelagem do Banco de Dados:**
   * Identificar e definir as entidades e seus relacionamentos com base nos requisitos do sistema.
   * Elaborar o modelo Entidade-Relacionamento, incluindo as tabelas principais (Fato Atendimento) e de dimensão (Cliente, Atendente, Motivo, Canal de Atendimento, Produto ou Serviço).
   * Garantir a correta definição das chaves primárias e estrangeiras para manter a integridade referencial.
2. **Escolhas de Campos e Tipos de Dados:**
   * Determinar os tipos de dados apropriados para cada campo em todas as tabelas do banco de dados.
   * Assegurar a normalização do banco de dados para evitar redundância e inconsistência.
   * Validar a consistência dos dados e garantir a precisão das informações armazenadas.
3. **Otimização do Banco de Dados e Criação da Modelagem (Tabelas e Relacionamentos - Engenharia Reversa):**
   * Identificar oportunidades de otimização no banco de dados, como a criação de índices em campos frequentemente utilizados em consultas.
   * Implementar técnicas de particionamento de tabelas para melhorar o desempenho das consultas.
   * Escrever consultas SQL eficientes e otimizadas, evitando subconsultas desnecessárias e garantindo o uso adequado de índices.
4. **População do Banco e Desenvolvimento dos KPIs (Indicadores-Chave de Desempenho):**
   * Cada tabela deve ser populada com 100.000 (cem mil) linhas.
   * Popular cada uma das tabelas de acordo com o modelo de dados (colunas e tipo de dado de cada coluna) - 100 mil linhas cada tabela.
   * Identificar os KPIs relevantes para monitorar o desempenho do call center, como taxa de resolução, tempo médio de atendimento, taxa de abandono, etc.
   * Escrever consultas SQL para calcular e extrair os dados necessários para cada indicador-chave.
   * Justificar e explicar o funcionamento de cada consulta e explicar como os KPIs refletem a eficiência da operação do call center.
5. **Elaboração do Relatório (Dashboard) Gerencial:**
   * Nome do Responsável:
   * Criação do dashboard gerencial (Power BI ou Excel) e compilação do relatório escrito.
   * Compilar todas as informações coletadas ao longo do projeto em um relatório gerencial abrangente.
   * Incluir análises sobre as escolhas de campos, otimizações do banco de dados e desenvolvimento dos KPIs.
   * Apresentar os resultados de forma clara e objetiva, destacando insights e recomendações para a gestão estratégica do call center.

**PROJETO DE OTIMIZAÇÃO DE BANCO DE DADOS DE CALL CENTER**

**Introdução**

Este documento aprofunda a análise dos recursos de otimização utilizados no modelo Star Schema apresentado, detalhando as escolhas de tabelas, tipos de dados e índices de consulta, com o objetivo de aprimorar o desempenho e a eficiência do banco de dados.

**Análise das Tabelas**

O modelo Star Schema apresenta um design adequado para consultas analíticas complexas, com tabelas dimensionais e de fatos bem estruturadas. As tabelas dimensionais armazenam atributos descritivos dos dados de fato, enquanto a tabela de fatos centraliza os dados quantitativos e as chaves primárias para referenciar as dimensões.

**Detalhamento de Tabelas:**

1. **TB\_CLIENTE**

A primeira tabela que temos é a tabela de Clientes. Ela será responsável por armazenar os dados cadastrais dos clientes que serão atendidos pelo nosso callcenter.

- Os dados que armazenaremos serão:

SEQCLIENTE: código sequencial único do tipo inteiro que identificará o cliente. É a chave primária da tabela.

NOME: campo do tipo varchar para guardar o nome completo do cliente.

EMAIL: campo do tipo varchar para armazenar o email do cliente.

TELEFONE: campo do tipo varchar para armazenar o telefone do cliente (limitado a 14 caracteres).

CPF: Campo do tipo varchar para armazenar o CPF do cliente (limitado a 11 caracteres).

DATA\_CADASTRO: campo do tipo DATETIME para que seja utilizado para armazenar a data em que o cliente foi cadastrado.

1. **TB\_DEPARTAMENTO**

A segunda tabela é a de Departamento, que fará o registro de todas as áreas que temos no nosso callcenter. Ela é importante pois servirá de base para cadastrarmos nossos funcionários posteriormente e associarmos a um departamento.

- Os dados que armazenaremos nessa tabela serão

SEQDEPARTAMENTO: código sequencial único do tipo inteiro que identificará o departamento. É a chave primária da tabela.

NOME\_DEPARTAMENT: campo do tipo varchar para armazenar o nome do departamento.

DATA\_CRIAÇÃO: campo do tipo datetime para armazenar a data em que o departamento foi criado.

1. **TB\_ATENDENTE**

A próxima tabela é a de atendentes, que será utilizada para armazenar os dados dos atendentes do callcenter. Ela é criada posteriormente pois é a primeira tabela onde iremos referenciar um registro de outra tabela.

- Os dados que armazenaremos nessa tabela serão

SEQATENDENTE: código sequencial único do tipo inteiro que identificará o atendente. É a chave primária da tabela.

NOME\_ATENDENTE: campo do tipo varchar para guardar o nome completo do atendente.

CARGO: campo do tipo varchar para armazenar o cargo do atendente.

SEQDEPARTAMENTO: campo do tipo inteiro, responsável por conectar um atendente a um departamento. É uma chave estrangeira que se conecta ao SEQDEPARTAMENTO da tabela TB\_DEPARTAMENTO.

NIVEL\_EXPERIENCIA: campo do tipo varchar para armazenar o nível de senioridade do atendente. Exemplo: Assistente, Júnior, Pleno, Sênior.

DATA\_ADMISSAO: campo do tipo datetime para armazenar a data em que o atendente foi contratado pela empresa.

DATA\_ALTERACAO: campo do tipo datetime para armazenar a data da ultima alteração que foi realizada no cadastro do atendente. Exemplo: o atendente subiu de Júnior para Pleno; com isso, é preciso registrar nesse campo a data em que foi alterado esse campo.

1. **TB\_MOTIVO\_ATENDIMENTO**

Essa tabela é responsável por armazenar os motivos pelo qual um cliente possa a vir a contactar o nosso callcenter. Nela, também teremos uma conexão com a tabela de departamento, para fazer com que cada departamento seja responsável por atender determinado tipo de problema.

- Os dados que armazenaremos nessa tabela serão

SEQMOTIVO: código sequencial único do tipo inteiro que identificará o motivo. É a chave primária da tabela.

NOME\_MOTIVO: campo do tipo varchar para guardar o motivo.

PRIORIDADE: campo do tipo inteiro para armazenar, para identificar de 1 a 10 qual o nível de prioridade que aquele problema deve ter no atendimento.

DATA\_CRIACAO: campo do tipo datetime para armazenar a data em que o motivo foi cadastrado.

DATA\_ALTERACAO: campo do tipo datetime para armazenar a data em que o motivo foi alterado pela ultima vez.

DEPARTAMENTO\_RESPONSAVEL: campo do tipo inteiro, responsável por conectar um motivo a um departamento. É uma chave estrangeira que se conecta ao SEQDEPARTAMENTO da tabela TB\_DEPARTAMENTO.

1. **TB\_PRODUTO**

Essa é a tabela responsável por armazenar os dados dos produtos que nós atendemos no nosso callcenter. Ela também possui uma ligação com a tabela de DEPARTAMENTO, para que departamentos específicos possam ficar responsáveis pelos produtos específicos.

- Os dados que armazenaremos nessa tabela serão:

SEQPRODUTO: código sequencial único do tipo inteiro que identificará o produto. É a chave primária da tabela.

NOME\_PRODUTO: campo do tipo varchar para armazenar o nome do produto.

DESCRICAO\_PRODUTO: campo do tipo varchar para armazenar uma informação mais detalhada ou complementar do produto.

PRECO: campo do tipo decimal para armazenar o preço do produto.

DEPARTAMENTO\_RESPONSAVEL: campo do tipo inteiro para armazenar o código do departamento responsável pelo produto. É uma chave estrangeira que se conecta ao SEQDEPARTAMENTO da tabela TB\_DEPARTAMENTO.

DATA\_CRIACAO: campo do tipo datetime para armazenar a data em que o produto foi cadastrado.

DATA\_ALTERACAO: campo do tipo datetime para armazenar a data em que o produto foi alterado pela ultima vez.

STATUS\_PRODUTO: campo do tipo char para armazenar se o produto está ativo ou não. As opções são: "A" para ativo "I" para inativo.

1. **TB\_CANALATENDIMENTO**

Essa tabela é responsável por armazenar os dados dos tipos de canais pelo qual os clientes podem entrar em contato com o nosso callcenter. Exemplo: Whatsapp, Email, Telefone, Telegram, Chatbot.

- Os dados que armazenaremos nessa tabela serão:

SEQCANAL: código sequencial único do tipo inteiro que identificará o canal de atendimento. É a chave primária da tabela.

NOME\_CANAL: campo do tipo varchar para armazenar o nome do canal de atendimento.

INFORMACOES\_CANAL: campo do tipo varchar para armazenar dados complementares ou detalhadas do canal.

TEMPO\_RESPOSTA: campo do tipo time para armazenar o quanto deve ser o tempo máximo de resposta do atendimento para o canal em questão.

DISPONIBILIDADE: campo do tipo varchar para armazenar a disponibilidade que o canal de atendimento ficará de prontidão. Exemplo: 24 horas, horário comercial, das 08 as 18, etc.

STATUS\_CANAL: campo do tipo char para armazenar se o canal está ativo ou não. As opções são: "A" para ativo "I" para inativo.

1. **TB\_ATENDIMENTO**

Essa é a principal tabela do nosso sistema de callcenter. Ela é responsável por armazenar os dados de cada atendimento que realizamos. Ela também possui uma referência para as tabelas de cliente, atendente, motivo, produto e canal. Todos esses campos são referenciados pelos seus códigos sequenciais, afim de não armazenarmos dados redundantes na tabela (como nome do cliente, nome do atendente) e garantir que a tabela tenha a melhor performance possível.

- Os dados que armazenaremos nessa tabela serão:

SEQATENDIMENTO: código sequencial único do tipo inteiro que identificará o atendimento realizado. É a chave primária da tabela.

DATA\_HORA\_INICIO\_ATENDIMENTO: campo do tipo datetime para armazenar o dia e horário em que o atendimento iniciou.

DATA\_HORA\_FIM\_ATENDIMENTO: campo do tipo datetime para armazenar o dia e horário em que o atendimento foi finalizado.

SEQCLIENTE: campo do tipo inteiro que serve para referenciar o cliente que foi atendido. É uma chave estrangeira, que se liga a tabela TB\_CLIENTE através do campo SEQCLIENTE.

SEQATENDENTE: campo do tipo inteiro que serve para referenciar o atendente que realizou o serviço de atendimento. É uma chave estrangeira que se liga a tabela TB\_ATENDENTE através do campo SEQATENDENTE.

SEQMOTIVO: campo do tipo inteiro que serve para referenciar o motivo do atendimento. É uma chave estrangeira que se liga a tabela TB\_MOTIVO\_ATENDIMENTO através do campo SEQMOTIVO.

PROTOCOLO: campo do tipo varchar responsável por armazenar o código de protocolo de atendimento que o cliente pode utilizar para referenciar o atendimento.

SEQPRODUTO: campo do tipo inteiro que serve para referenciar o produto no qual estamos prestando atendimento. É uma chave estrangeira que se liga a tabela TB\_PRODUTO através do campo SEQPRODUTO.

SEQCANAL: campo do tipo inteiro que serve para referenciar por qual canal de atendimento o cliente entrou em contato com o callcenter. É uma chave estrangeira que se liga a tabela TB\_CANALATENDIMENTO através do campo SEQCANAL.

ENPS: campo do tipo tinyint que serve para armazenar a nota que foi dada ao atendimento pelo cliente.

**Detalhamento dos índices:**

Os índices foram selecionados e criados todos com base no mesmo princípio: ser utilizado nos principais campos nos quais as tabelas farão conexão umas com as outras e serão utilizadas nas cláusulas where.

1. **Tabela TB\_ATENDIMENTO:**

Na tabela de atendimento, foram escolhidos os campos SEQCLIENTE, SEQATENDENTE, SEQMOTIVO, SEQPRODUTO e SEQCANAL pois todos eles são as chaves estrangeiras da tabela, portanto, ao realizar um JOIN, esses campos serão primordiais para a melhorar a performance nas consultas.

CREATE INDEX IDX\_ATENDIMENTO\_SEQCLIENTE ON TB\_ATENDIMENTO(SEQCLIENTE);

CREATE INDEX IDX\_ATENDIMENTO\_SEQATENDENTE ON TB\_ATENDIMENTO(SEQATENDENTE);

CREATE INDEX IDX\_ATENDIMENTO\_SEQMOTIVO ON TB\_ATENDIMENTO(SEQMOTIVO);

CREATE INDEX IDX\_ATENDIMENTO\_SEQPRODUTO ON TB\_ATENDIMENTO(SEQPRODUTO);

CREATE INDEX IDX\_ATENDIMENTO\_SEQCANAL ON TB\_ATENDIMENTO(SEQCANAL);

1. **Tabela TB\_CANALATENTIMENTO:**

Na tabela de canal de atendimento, foi criado no campo SEQCANAL, devido ao mesmo ser utilizado para se conectar com a tabela de atendimento.

CREATE INDEX IDX\_CANAL\_SEQ ON TB\_CANALATENDIMENTO (SEQCANAL);

1. **Tabela TB\_CLIENTE**

Na tabela de Cliente, foram criados índices tanto no campo de SEQCLIENTE quando no campo de CPF, pois apesar do SEQCLIENTE ser o principal campo responsável por unir as informações do cliente a outras tabelas, o campo CPF será o mais utilizado para identificar um cliente específico em alguma consulta.

CREATE INDEX IDX\_CLIENTE\_CPF ON TB\_CLIENTE(CPF);

CREATE INDEX IDX\_CLIENTE\_SEQ ON TB\_CLIENTE(SEQCLIENTE);

1. **Tabela TB\_ATENDENTE**

Na tabela de atendente, foram criados índices tanto na chave primária, quanto na chave estrangeira.

CREATE INDEX IDX\_ATENDENTE\_SEQ ON TB\_ATENDENTE(SEQATENDENTE);

CREATE INDEX IDX\_ATENDENTE\_DPTO ON TB\_ATENDENTE(SEQDEPARTAMENTO);

1. **TB\_MOTIVO\_ATENDIMENTO**

Na tabela de motivo atendimento, foram criados índices para a chave primária, juntamente com seu nome e para a chave estrangeira.

CREATE INDEX IDX\_MOTIVO\_SEQ ON TB\_MOTIVO\_ATENDIMENTO(SEQMOTIVO, NOME\_MOTIVO);

CREATE INDEX IDX\_MOTIVO\_DPTO ON TB\_MOTIVO\_ATENDIMENTO(DEPARTAMENTO\_RESPONSAVEL);

1. **Tabela TB\_PRODUTO**

Na tabela de produto, foi criado o índice para a chave primária, juntamente com a chave estrangeira de departamento.

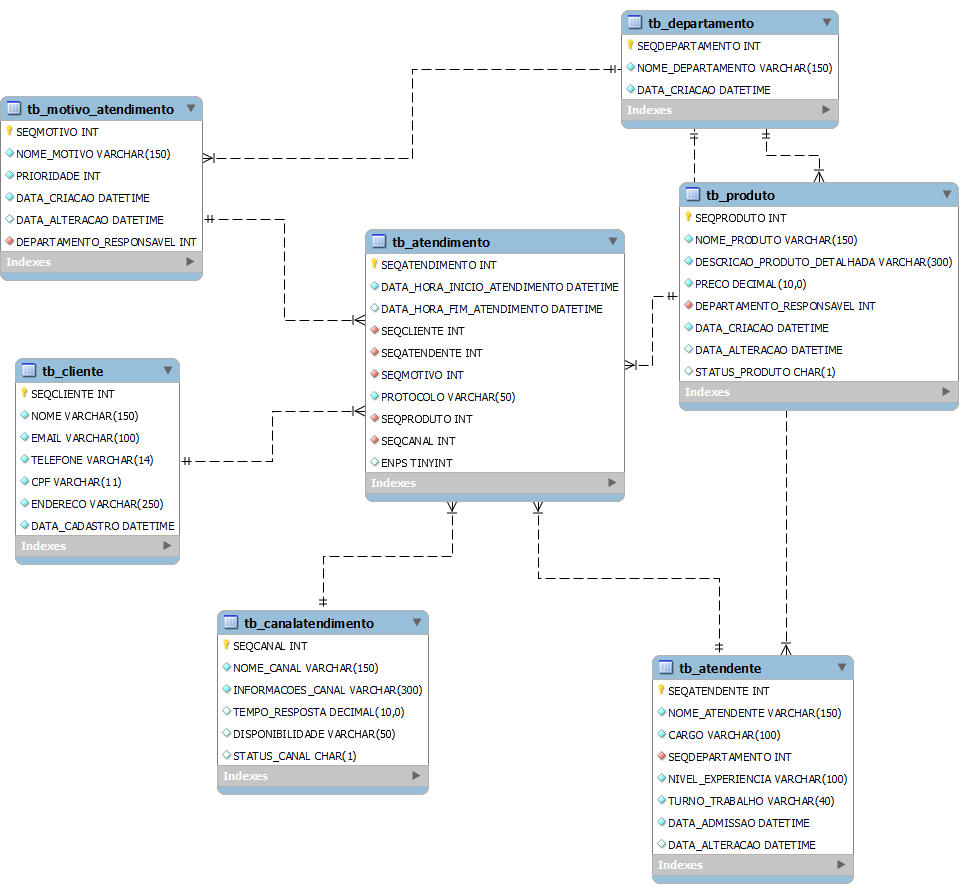
CREATE INDEX IDX\_PRODUTO\_SEQ ON TB\_PRODUTO(SEQPRODUTO, DEPARTAMENTO\_RESPONSAVEL);

1. **Tabela TB\_DEPARTAMENTO**

Na tabela de departamento, foi criado o índice tanto para a chave primária quanto para o campo do nome, que provavelmente será o mais utilizado para identificar o departamento numa consulta na tabela.

CREATE INDEX IDX\_DPTO ON TB\_DEPARTAMENTO (SEQDEPARTAMENTO, NOME\_DEPARTAMENTO);

**ESTRATÉGIA DETALHADA DE OTIMIZAÇÃO NO MODELO STAR SCHEMA**

A estratégia de otimização implementada no modelo Star Schema, abrangendo a modelagem das tabelas, a escolha dos tipos de dados e a utilização de recursos de índices. O objetivo é garantir a eficiência do armazenamento, a precisão das informações e o bom desempenho do banco de dados, atendendo às necessidades de análise de dados complexas e oferecendo uma experiência de consulta rápida e confiável.  
  
  


**1. Modelagem de Tabelas:**

* **Modelo Star Schema:**
  + Adotado para otimizar consultas analíticas complexas, com tabelas dimensionais e de fatos bem estruturadas.
  + Tabelas dimensionais armazenam atributos descritivos dos dados de fato, enquanto a tabela de fatos centraliza os dados quantitativos e as chaves primárias para referenciar as dimensões.
  + Essa estrutura facilita a agregação e análise de dados de diferentes dimensões, otimizando o desempenho das consultas.
* **Normalização Estratégica:**
  + Nível de normalização 3NF (Terceira Forma Normal) implementado para minimizar redundância de dados e garantir integridade referencial.
  + Evita anomalias de atualização e inserção, além de otimizar o espaço de armazenamento.
  + As tabelas dimensionais armazenam apenas atributos relevantes para as análises, enquanto a tabela de fatos contém os dados quantitativos e as chaves primárias para referenciar as dimensões.
* **Chaves Estrangeiras:**
  + Utilizadas para conectar as tabelas dimensionais à tabela de fatos, garantindo a integridade referencial.
  + Permitem a navegabilidade eficiente entre as tabelas e facilitam a recuperação de dados relacionados.
  + As chaves estrangeiras são implementadas com índices para otimizar as junções entre as tabelas.

**2. Tipos de Dados:**

* **Seleção Estratégica:**
  + A escolha dos tipos de dados foi feita com base na natureza dos dados, nas necessidades de armazenamento e nas operações que serão realizadas sobre eles.
  + Priorizada a eficiência do armazenamento, a precisão das informações e o desempenho das consultas.
* **Tipos de Dados Utilizados:**
  + **INT:**
    - Utilizado para campos que representam valores numéricos inteiros e únicos, como identificadores de entidades (SEQCLIENTE, SEQDEPARTAMENTO, etc.).
    - Permite ordenação rápida, indexação eficiente e consultas otimizadas.
  + **VARCHAR:**
    - Utilizado para campos que armazenam strings com tamanhos variáveis, como nomes, descrições e informações textuais (NOME, NOME\_DEPARTAMENTO, etc.).
    - Oferece flexibilidade para diferentes tamanhos de dados e otimiza o espaço de armazenamento.
  + **DATETIME:**
    - Utilizado para campos que armazenam datas e horas com precisão, como datas de cadastro, datas de atendimento e datas de criação (DATA\_CADASTRO, DATA\_HORA\_INICIO\_ATENDIMENTO, etc.).
    - Permite ordenação cronológica, filtragem por data e análise de tendências.
  + **DECIMAL:**
    - Utilizado para campos que representam valores monetários, como preços de produtos (PRECO).
    - Garante precisão nos cálculos financeiros e na análise de dados monetários.
  + **TINYINT:**
    - Utilizado para campos que armazenam valores numéricos pequenos com pouca variação, como notas de avaliação (ENPS).
    - Otimiza o espaço de armazenamento e permite consultas eficientes.
* **Justificativa da Escolha:**
  + A escolha dos tipos de dados foi feita com o objetivo de garantir:
    - Eficiência no armazenamento: Os tipos de dados escolhidos otimizam o espaço de armazenamento sem comprometer a precisão das informações.
    - Precisão das informações: Os tipos de dados garantem que os dados sejam armazenados e processados de forma precisa, atendendo às necessidades de análise.
    - Desempenho das consultas: Os tipos de dados e os índices otimizam as consultas, permitindo que sejam realizadas de forma rápida e eficiente.

**3. Índices:**

* **Criados em Campos Chave:**
  + Índices criados nas chaves primárias de todas as tabelas para otimizar a recuperação de registros específicos.
  + Permitem acesso rápido aos dados e garantem o bom desempenho das consultas que utilizam as chaves primárias como critério de busca.
* **Criados em Campos de Consulta Frequente:**
  + Índices criados em campos que são frequentemente utilizados como critérios de filtro ou ordenação nas consultas.

Otimizam a performance das consultas que utilizam esses campos, ermitindo a filtragem e ordenação eficientes dos dados.

* **Justificativa da Utilização:**
  + A utilização de índices melhora o desempenho das consultas ao fornecer um ponteiro rápido para os dados relevantes.
  + Ao invés de ler toda a tabela, o banco de dados pode utilizar o índice para localizar rapidamente os registros que correspondem ao critério de busca.

A estratégia de otimização implementada no modelo Star Schema, aliada à modelagem de tabelas, à seleção de tipos de dados e à utilização de índices, garante a eficiência do armazenamento, a precisão das informações e o bom desempenho geral do modelo. Essa abordagem permite realizar consultas analíticas complexas de forma rápida e confiável, atendendo às necessidades de análise de dados e oferecendo uma experiência satisfatória para os usuários do sistema.

**DASHBOARD PARA ANÁLISE DETALHADA DO ATENDIMENTO AO CLIENTE**

Foi elaborado um dashboard no Power BI para avaliar indicadores de desempenho crucial no atendimento ao cliente. A estratégia de otimização utilizada no modelo Star Schema garante a eficiência do armazenamento, a precisão das informações e o bom desempenho geral do modelo, permitindo a construção de visualizações poderosas e interativas.

**Visualizações Estratégicas:**

O dashboard apresenta uma série de gráficos selecionados para fornecer uma visão abrangente do desempenho do atendimento ao cliente:

**1. Tempo Médio de Atendimento por Nome do Produto:**

* **Tipo de Gráfico:** Coluna ou Barra
* **Análise:** Identifica os produtos que exigem maior tempo de atendimento, permitindo ações direcionadas para otimizar processos e reduzir o tempo de resolução.

**2. Tempo Médio de Atendimento por Nome do Canal:**

* **Tipo de Gráfico:** Coluna ou Barra
* **Análise:** Avalia a eficiência de cada canal de atendimento, possibilitando identificar gargalos e direcionar investimentos para os canais que apresentam maior demanda.

**3. Tempo Médio de Atendimento por Nome do Departamento:**

* **Tipo de Gráfico:** Coluna ou Barra
* **Análise:** Mede a performance de cada departamento em relação à resolução de atendimentos, auxiliando na identificação de áreas que necessitam de treinamento ou otimização de processos.

**4. Tempo Médio de Atendimento por Nome do Atendente:**

* **Tipo de Gráfico:** Coluna ou Barra
* **Análise:** Monitora o desempenho individual dos atendentes, permitindo identificar talentos, oferecer treinamento direcionado e otimizar a alocação de recursos.

**5. TMA por Motivo de Atendimento:**

* **Tipo de Gráfico:** Coluna
* **Análise:** Compreende o tempo médio de atendimento para cada motivo, possibilitando identificar padrões, investigar causas raízes e implementar soluções para os motivos que geram maior tempo de resolução.

**6. Total de Atendimentos e Mediana da Taxa de Atendimento por Mês:**

* **Tipo de Gráfico:** Linha e coluna
* **Análise:** Visualiza o volume de atendimentos e a taxa de resolução ao longo do tempo, permitindo identificar tendências, sazonalidades e áreas que precisam de atenção.

**7. Soma do Tempo de Resolução por Nome do Canal:**

* **Tipo de Gráfico:** Coluna ou Barra
* **Análise:** Quantifica o tempo total de atendimento por canal, fornecendo insights sobre a carga de trabalho e a necessidade de otimização de recursos.

**Importação e Relacionamento de Tabelas:**

A modelagem Star Schema otimizada foi importada para o Power BI através do método de importação do MySQL, garantindo a integridade e a confiabilidade dos dados. As tabelas foram relacionadas de acordo com a lógica dimensional, permitindo a criação de relacionamentos eficientes e a construção de visualizações precisas.

**KPIs Estratégicos:**

Para complementar a análise, foram inseridos KPIs estratégicos de desenvolvimento, detalhados a seguir:

**Query 1 - Quantidade\_Atendimentos\_Mês\_Atendente:**

* **Objetivo:** Quantificar o número de atendimentos realizados por cada atendente em um determinado mês.
* **Utilização:** Monitorar a produtividade individual, identificar gargalos e direcionar treinamentos.  
    
  # KPI DE NOME ATENDENTE, MÊS, ANO E NÚMERO DE ATENDIMENTOS POR MÊS.

SET lc\_time\_names='pt\_PT';

CREATE VIEW VW\_RELATORIO\_DIARIO AS

SELECT

AT.NOME\_ATENDENTE,

MONTHNAME(A.DATA\_HORA\_INICIO\_ATENDIMENTO) AS MES,

YEAR(A.DATA\_HORA\_INICIO\_ATENDIMENTO) AS ANO,

COUNT(A.SEQATENDIMENTO) AS NUMERO\_ATENDIMENTOS

FROM

TB\_ATENDIMENTO A

INNER JOIN TB\_ATENDENTE AT ON A.SEQATENDENTE = AT.SEQATENDENTE

GROUP BY

AT.NOME\_ATENDENTE,

MONTHNAME(A.DATA\_HORA\_INICIO\_ATENDIMENTO),

YEAR(A.DATA\_HORA\_INICIO\_ATENDIMENTO);

**Query 2 - Tempo\_Médio\_Resolução\_Motivo:**

* **Objetivo:** Calcular o tempo médio de resolução para cada motivo de atendimento.
* **Utilização:** Identificar os motivos que geram maior tempo de resolução e implementar ações para otimizar os processos.

SET lc\_time\_names='pt\_PT';

SELECT

C.NOME\_CANAL,

M.NOME\_MOTIVO,

SEC\_TO\_TIME(ROUND(AVG(TIMESTAMPDIFF(SECOND, A.DATA\_HORA\_INICIO\_ATENDIMENTO, A.DATA\_HORA\_FIM\_ATENDIMENTO)) / 3600)) AS TEMPO\_MEDIO\_RESOLUCAO\_HORAS

FROM

TB\_ATENDIMENTO A

INNER JOIN TB\_CANALATENDIMENTO C ON A.SEQCANAL = C.SEQCANAL

INNER JOIN TB\_MOTIVO\_ATENDIMENTO M ON A.SEQMOTIVO = M.SEQMOTIVO

WHERE

A.DATA\_HORA\_FIM\_ATENDIMENTO IS NOT NULL

GROUP BY

C.NOME\_CANAL,

M.NOME\_MOTIVO

ORDER BY

TEMPO\_MEDIO\_RESOLUCAO\_HORAS DESC;

**Query 3 - Taxa\_Resolução\_Atendimento:**

* **Objetivo:** Determinar a taxa de resolução de cada atendimento, considerando se o problema do cliente foi solucionado na primeira interação.
* **Utilização:** Avaliar a eficiência do atendimento à primeira vista, identificar áreas de melhoria e direcionar treinamentos para os atendentes.

SELECT

D.NOME\_DEPARTAMENTO,

MONTH(A.DATA\_HORA\_INICIO\_ATENDIMENTO) AS MES,

YEAR(A.DATA\_HORA\_INICIO\_ATENDIMENTO) AS ANO,

COUNT(A.SEQATENDIMENTO) AS TOTAL\_ATENDIMENTOS,

COUNT(CASE WHEN A.DATA\_HORA\_FIM\_ATENDIMENTO IS NOT NULL THEN 1 END) AS ATENDIMENTOS\_RESOLVIDOS,

ROUND((COUNT(CASE WHEN A.DATA\_HORA\_FIM\_ATENDIMENTO IS NOT NULL THEN 1 END) / COUNT(A.SEQATENDIMENTO)) \* 100, 2) AS TAXA\_RESOLUCAO

FROM

TB\_ATENDIMENTO A

INNER JOIN TB\_ATENDENTE AT ON A.SEQATENDENTE = AT.SEQATENDENTE

INNER JOIN TB\_DEPARTAMENTO D ON AT.SEQDEPARTAMENTO = D.SEQDEPARTAMENTO

GROUP BY

D.NOME\_DEPARTAMENTO,

MONTH(A.DATA\_HORA\_INICIO\_ATENDIMENTO),

YEAR(A.DATA\_HORA\_INICIO\_ATENDIMENTO);

**Query 4 - Tempo\_Medio\_Resolução\_2023:**

* **Objetivo:** Calcular o tempo médio de resolução de atendimentos no ano de 2023.
* **Utilização:** Analisar a performance geral do atendimento ao cliente em 2023 e identificar tendências ao longo do ano.

SELECT

M.SEQMOTIVO,

M.NOME\_MOTIVO,

SEC\_TO\_TIME(ROUND(AVG(TIME\_TO\_SEC(TEMPO\_MEDIO\_RESOLUCAO\_HORAS)))) AS MEDIA\_TEMPO\_MEDIO\_RESOLUCAO\_HORAS

FROM (

SELECT

A.SEQMOTIVO,

SEC\_TO\_TIME(ROUND(AVG(TIMESTAMPDIFF(SECOND, A.DATA\_HORA\_INICIO\_ATENDIMENTO, A.DATA\_HORA\_FIM\_ATENDIMENTO)) / 3600)) AS TEMPO\_MEDIO\_RESOLUCAO\_HORAS

FROM

TB\_ATENDIMENTO A

WHERE

A.DATA\_HORA\_FIM\_ATENDIMENTO IS NOT NULL

AND YEAR(A.DATA\_HORA\_INICIO\_ATENDIMENTO) = 2023

GROUP BY

A.SEQMOTIVO

) AS subquery

INNER JOIN TB\_MOTIVO\_ATENDIMENTO M ON subquery.SEQMOTIVO = M.SEQMOTIVO

GROUP BY

M.SEQMOTIVO, M.NOME\_MOTIVO;

**Query 5 - Tempo\_Medio\_Resolução\_2024:**

* **Objetivo:** Calcular o tempo médio de resolução de atendimentos no ano de 2024 (até o momento da consulta).
* **Utilização:** Monitorar a performance do atendimento ao cliente em 20

SELECT

M.SEQMOTIVO,

M.NOME\_MOTIVO,

SEC\_TO\_TIME(ROUND(AVG(TIME\_TO\_SEC(TEMPO\_MEDIO\_RESOLUCAO\_HORAS)))) AS MEDIA\_TEMPO\_MEDIO\_RESOLUCAO\_HORAS

FROM (

SELECT

A.SEQMOTIVO,

SEC\_TO\_TIME(ROUND(AVG(TIMESTAMPDIFF(SECOND, A.DATA\_HORA\_INICIO\_ATENDIMENTO, A.DATA\_HORA\_FIM\_ATENDIMENTO)) / 3600)) AS TEMPO\_MEDIO\_RESOLUCAO\_HORAS

FROM

TB\_ATENDIMENTO A

WHERE

A.DATA\_HORA\_FIM\_ATENDIMENTO IS NOT NULL

AND YEAR(A.DATA\_HORA\_INICIO\_ATENDIMENTO) = 2024

GROUP BY

A.SEQMOTIVO

) AS subquery

INNER JOIN TB\_MOTIVO\_ATENDIMENTO M ON subquery.SEQMOTIVO = M.SEQMOTIVO

GROUP BY

M.SEQMOTIVO, M.NOME\_MOTIVO;

**IMAGENS DO DASHBOARD**  
  
Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário

Descrição gerada automaticamente

## REPOSITÓRIO GITHUB COM DOCUMENTAÇÃO COMPLETA

Para facilitar o acesso à documentação completa do projeto, foi criado um repositório público no GitHub, disponível no seguinte link:

<https://github.com/jacksonwsup/bancodedados>

**Conteúdo do Repositório:**

O repositório contém os seguintes arquivos importantes para a avaliação do projeto:

**Scripts SQL:**

* CALLCENTER\_TB\_ATENDENTE.sql: Cria a tabela TB\_ATENDENTE no banco de dados.
* CALLCENTER\_TB\_ATENDIMENTO.sql: Cria a tabela TB\_ATENDIMENTO no banco de dados.
* CALLCENTER\_TB\_CANALATENDIMENTO.sql: Cria a tabela TB\_CANALATENDIMENTO no banco de dados.
* CALLCENTER\_TB\_CLIENTE.sql: Cria a tabela TB\_CLIENTE no banco de dados.
* CALLCENTER\_TB\_DEPARTAMENTO.sql: Cria a tabela TB\_DEPARTAMENTO no banco de dados.
* CALLCENTER\_TB\_MOTIVO\_ATENDIMENTO.sql: Cria a tabela TB\_MOTIVO\_ATENDIMENTO no banco de dados.
* CALLCENTER\_TB\_PRODUTO.sql: Cria a tabela TB\_PRODUTO no banco de dados.
* CALLCENTER\_routines.sql: Cria rotinas SQL para auxiliar na manipulação dos dados.
* Create\_Database.sql: Cria o banco de dados CALLCENTER.
* Create\_Index.sql: Cria índices nas tabelas do banco de dados para otimizar as consultas.

**Diagrama:**

* DiagramaV2.png: Diagrama UML representando o modelo de dados Star Schema.

**Dump de Dados:**

* Dump20240504-1.rar: Arquivo compactado contendo o dump completo do banco de dados em 04 de maio de 2024.

**Power BI:**

* PROJETO DE OTIMIZAÇÃO DASHBOARD DE INDICADORES.pbix: Arquivo do projeto Power BI contendo o dashboard de indicadores de desempenho.

**Scripts para Geração de Relatórios:**

* PROJETO DE OTIMIZAÇÃO.sql: Script SQL para gerar os relatórios Kpis.
* README.md: Arquivo de texto contendo informações sobre o projeto e instruções de uso.

**Relatórios SQL:**

* relatorio\_kpi\_Atendimentos.sql: Script SQL para gerar o relatório KPI Atendimentos.
* relatorio\_kpi\_canalatendimento\_motivos.sql: Script SQL para gerar o relatório KPI Canal Atendimento x Motivos.
* relatorio\_kpi\_media\_tempo\_atendimentos\_por\_motivos.sql: Script SQL para gerar o relatório KPI Tempo Médio Atendimento por Motivo.