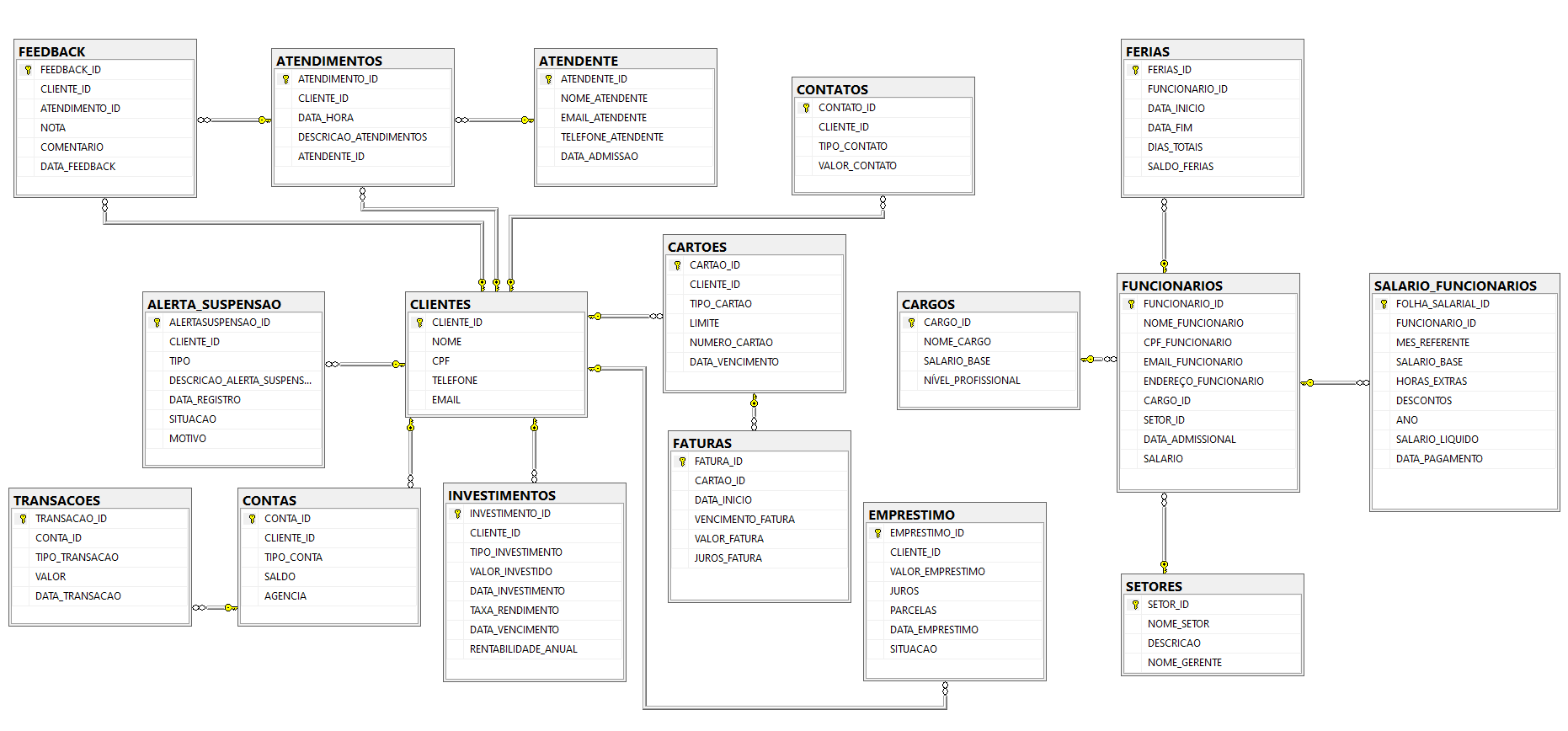
**Objetivo:**

Melhorar a estrutura do banco de dados, remover a tabela FUNCIONARIOS e suas dependências, e migrar o esquema para MySQL.

# Modelo Conceitual Antigo (SQL Server):

O modelo conceitual original do banco de dados DB\_BancoInter no SQL Server era composto pelas seguintes entidades (tabelas) e seus relacionamentos principais:

O modelo conceitual original representava um sistema bancário com entidades como CLIENTES, CONTAS, TRANSACOES, CONTATOS, EMPRESTIMO, CARTOES, FATURAS, ALERTA\_SUSPENSAO e INVESTIMENTOS, focadas nas operações bancárias com os clientes.

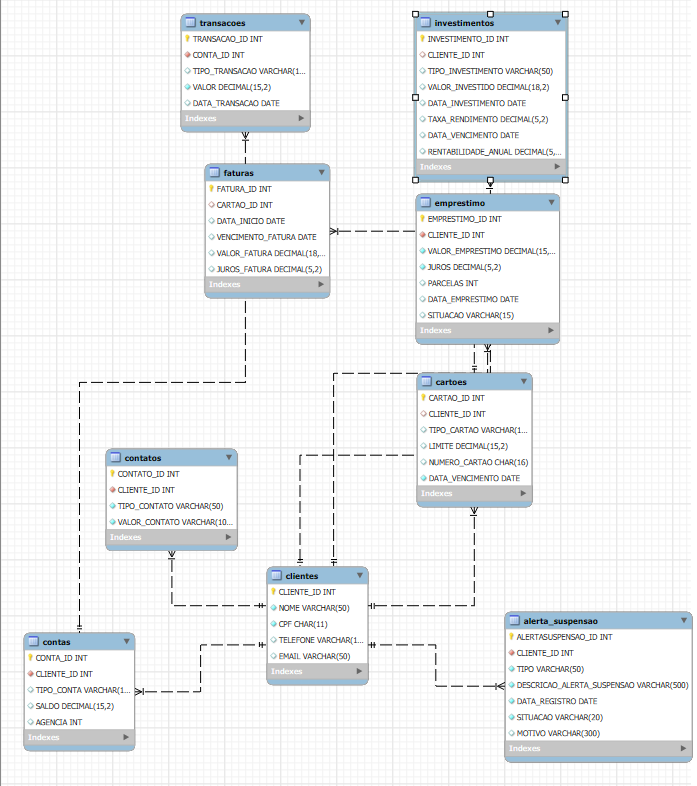
Adicionalmente, o modelo incluía entidades relacionadas à gestão interna do banco, como FUNCIONARIOS, ATENDENTE, CARGOS, SETORES, ATENDIMENTOS, FEEDBACK, SALARIO\_FUNCIONARIOS e FERIAS. Estas tabelas permitiam o registro de informações sobre os funcionários, seus cargos, setores, os atendimentos realizados aos clientes, o feedback desses atendimentos e informações sobre salários e férias dos funcionários.

Os relacionamentos indicavam como essas entidades estavam interligadas, por exemplo:

* Um CLIENTE poderia ter várias CONTAS e vários CONTATOS, EMPRESTIMOS, CARTOES, ALERTAS\_SUSPENSAO e INVESTIMENTOS.
* Uma CONTA poderia ter várias TRANSACOES.
* Um CARTAO poderia ter várias FATURAS.
* Um CLIENTE poderia ter vários ATENDIMENTOS, que eram realizados por um ATENDENTE.
* Um ATENDENTE estava associado a um FUNCIONARIO.
* Um FUNCIONARIO pertencia a um CARGO e a um SETOR, e possuía registros em SALARIO\_FUNCIONARIOS e FERIAS.
* Um ATENDIMENTO poderia receber vários FEEDBACKS.

**Modelo Conceitual Novo (MySQL):**

Após a remoção das tabelas relacionadas a funcionários e a migração para MySQL, o novo modelo conceitual do banco de dados DB\_BancoInter é composto pelas seguintes entidades e seus relacionamentos principais:



**Legenda:**

* PK: Primary Key (Chave Primária)
* FK: Foreign Key (Chave Estrangeira)
* UN: Unique (Restrição de Unicidade)
* 1..\*: Relacionamento de um para muitos

**Descrição do Modelo Conceitual Novo:**

O novo modelo conceitual, agora implementado no MySQL, concentra-se nas entidades diretamente relacionadas aos clientes e suas operações bancárias. As tabelas e relacionamentos principais são:

* **CLIENTES:** Armazena informações básicas dos clientes.
* **CONTAS:** Contém detalhes das contas bancárias dos clientes.
* **TRANSACOES:** Registra as transações financeiras realizadas nas contas.
* **CONTATOS:** Mantém diferentes formas de contato dos clientes.
* **ALERTA\_SUSPENSAO:** Registra alertas e suspensões relacionadas às contas dos clientes.
* **EMPRESTIMO:** Armazena informações sobre os empréstimos concedidos aos clientes.
* **CARTOES:** Contém detalhes dos cartões (crédito e débito) dos clientes.
* **FATURAS:** Registra as faturas dos cartões de crédito.
* **INVESTIMENTOS:** Armazena informações sobre os investimentos realizados pelos clientes.

As tabelas relacionadas à gestão interna de funcionários (FUNCIONARIOS, ATENDENTE, CARGOS, SETORES, ATENDIMENTOS, FEEDBACK, SALARIO\_FUNCIONARIOS, FERIAS) foram removidas, simplificando o modelo de dados para focar exclusivamente nas informações dos clientes e suas atividades bancárias. Os relacionamentos de chave estrangeira garantem a integridade referencial entre as tabelas principais.

**Etapa 1: Análise e Sugestões de Melhoria no Esquema do SQL Server**

Antes da remoção e migração, foi realizada uma análise do esquema do banco de dados DB\_BancoInter no SQL Server fornecido. As seguintes observações e sugestões de melhoria foram identificadas:

* **Nomenclatura:** Considerada razoável, utilizando \_ para separação e prefixos/sufixos indicando o tipo (ID, NOME, etc.).
* **Tipos de Dados:** Pareceram adequados para as informações armazenadas.
* **Chaves Primárias:** Todas as tabelas possuem chaves primárias INT IDENTITY(1,1) para identificação única e autoincremento.
* **Chaves Estrangeiras:** Definidas corretamente para garantir a integridade referencial.
* **Restrições NOT NULL e UNIQUE:** Aplicadas de forma lógica para garantir a qualidade dos dados.
* **Restrições CHECK:** Presente na tabela FEEDBACK para validar a coluna NOTA.
* **Valores DEFAULT:** Utilizado na coluna SALDO da tabela CONTAS.

**Sugestões de Melhoria (não implementadas nesta etapa, mas para consideração futura no SQL Server):**

* **Padronização de VARCHAR:** Refinar o tamanho das colunas VARCHAR para melhor otimização do armazenamento.
* **Consistência em Nomes de FKs:** Adotar um padrão de nomenclatura mais uniforme para as chaves estrangeiras.
* **Índices:** Criar índices em colunas frequentemente utilizadas em consultas para melhorar o desempenho.

**Etapa 2: Remoção da Tabela FUNCIONARIOS e Tabelas Relacionadas (SQL Server)**

As seguintes tabelas e restrições de chave estrangeira foram removidas do banco de dados DB\_BancoInter no SQL Server:

1. **Remoção da Chave Estrangeira FK\_ATENDIMENTOS\_ATENDENTE na tabela ATENDIMENTOS:** SQL  
   ALTER TABLE ATENDIMENTOS

DROP CONSTRAINT FK\_ATENDIMENTOS\_ATENDENTE;

**Motivo:** A tabela ATENDIMENTOS possuía uma chave estrangeira referenciando a coluna ATENDENTE\_ID da tabela ATENDENTE. Como a tabela ATENDENTE estava relacionada à tabela FUNCIONARIOS (embora não diretamente por FK), e o objetivo era remover a funcionalidade relacionada a funcionários, essa restrição precisou ser removida para permitir a exclusão das tabelas subsequentes.

1. **Remoção da Chave Estrangeira FK\_FUNCIONARIOS\_FOLHADEPAGAMENTOS na tabela SALARIO\_FUNCIONARIOS:** SQL  
   ALTER TABLE SALARIO\_FUNCIONARIOS

DROP CONSTRAINT FK\_FUNCIONARIOS\_FOLHADEPAGAMENTOS;

**Motivo:** A tabela SALARIO\_FUNCIONARIOS possuía uma chave estrangeira referenciando a coluna FUNCIONARIO\_ID da tabela FUNCIONARIOS. Para remover a tabela FUNCIONARIOS, essa restrição precisou ser eliminada.

1. **Remoção da Chave Estrangeira FK\_FUNCIONARIOS\_FERIAS na tabela FERIAS:** SQL  
   ALTER TABLE FERIAS

DROP CONSTRAINT FK\_FUNCIONARIOS\_FERIAS;

**Motivo:** Similar à tabela SALARIO\_FUNCIONARIOS, a tabela FERIAS possuía uma chave estrangeira referenciando FUNCIONARIO\_ID.

1. **Remoção da tabela ATENDIMENTOS:** SQL  
   DROP TABLE ATENDIMENTOS;

**Motivo:** A tabela ATENDIMENTOS estava diretamente relacionada à tabela ATENDENTE, que por sua vez estava logicamente ligada à funcionalidade de funcionários. Além disso, a tabela FEEDBACK dependia de ATENDIMENTOS.

1. **Remoção da tabela FEEDBACK:** SQL  
   DROP TABLE FEEDBACK;

**Motivo:** A tabela FEEDBACK possuía uma chave estrangeira referenciando a coluna ATENDIMENTO\_ID da tabela ATENDIMENTOS, que foi removida.

1. **Remoção da tabela SALARIO\_FUNCIONARIOS:** SQL  
   DROP TABLE SALARIO\_FUNCIONARIOS;

**Motivo:** Esta tabela continha informações salariais dos funcionários e possuía uma chave estrangeira referenciando a tabela FUNCIONARIOS.

1. **Remoção da tabela FERIAS:** SQL  
   DROP TABLE FERIAS;

**Motivo:** Esta tabela armazenava informações sobre as férias dos funcionários e possuía uma chave estrangeira referenciando a tabela FUNCIONARIOS.

1. **Remoção da tabela FUNCIONARIOS:** SQL  
   DROP TABLE FUNCIONARIOS;

**Motivo:** Esta era a tabela principal a ser removida, conforme solicitado.

1. **Remoção da tabela CARGOS:** SQL  
   DROP TABLE CARGOS;

**Motivo:** A tabela CARGOS estava relacionada à tabela FUNCIONARIOS através de uma chave estrangeira. Com a remoção da tabela FUNCIONARIOS, presume-se que a tabela CARGOS não será mais necessária no novo escopo do banco de dados.

1. **Remoção da tabela SETORES:** SQL  
   DROP TABLE SETORES;

**Motivo:** Similar à tabela CARGOS, a tabela SETORES estava relacionada à tabela FUNCIONARIOS.

**Importante:** A execução desses comandos resultou na **perda permanente dos dados** contidos nas tabelas removidas.

**Etapa 3: Criação do Esquema no MySQL**

Um novo banco de dados chamado DB\_BancoInter foi criado no MySQL, e as seguintes tabelas foram criadas com suas respectivas estruturas, adaptadas para a sintaxe do MySQL:

SQL

CREATE DATABASE DB\_BancoInter;

USE DB\_BancoInter;

CREATE TABLE CLIENTES (

CLIENTE\_ID INT AUTO\_INCREMENT PRIMARY KEY,

NOME VARCHAR (50) NOT NULL,

CPF CHAR (11) NOT NULL UNIQUE,

TELEFONE VARCHAR (15),

EMAIL VARCHAR (50),

SENHA VARCHAR (50)

);

CREATE TABLE CONTAS(

CONTA\_ID INT AUTO\_INCREMENT PRIMARY KEY,

CLIENTE\_ID INT NOT NULL,

TIPO\_CONTA VARCHAR (10),

SALDO DECIMAL (15,2) DEFAULT 0.00,

AGENCIA INT,

FOREIGN KEY (CLIENTE\_ID) REFERENCES CLIENTES(CLIENTE\_ID)

);

CREATE TABLE CONTATOS (

CONTATO\_ID INT AUTO\_INCREMENT PRIMARY KEY,

CLIENTE\_ID INT NOT NULL,

TIPO\_CONTATO VARCHAR(50) NOT NULL,

VALOR\_CONTATO VARCHAR(100) NOT NULL,

FOREIGN KEY (CLIENTE\_ID) REFERENCES CLIENTES(CLIENTE\_ID)

);

CREATE TABLE ALERTA\_SUSPENSAO (

ALERTASUSPENSAO\_ID INT AUTO\_INCREMENT PRIMARY KEY,

CLIENTE\_ID INT NOT NULL,

TIPO VARCHAR(50) NOT NULL,

DESCRICAO\_ALERTA\_SUSPENSAO VARCHAR(500) NOT NULL,

DATA\_REGISTRO DATE NOT NULL,

SITUACAO VARCHAR(20) NOT NULL,

MOTIVO VARCHAR(300),

FOREIGN KEY (CLIENTE\_ID) REFERENCES CLIENTES(CLIENTE\_ID)

);

CREATE TABLE TRANSACOES (

TRANSACAO\_ID INT AUTO\_INCREMENT PRIMARY KEY,

CONTA\_ID INT NOT NULL,

TIPO\_TRANSACAO VARCHAR (15),

VALOR DECIMAL (15,2) NOT NULL,

DATA\_TRANSACAO DATE,

FOREIGN KEY (CONTA\_ID) REFERENCES CONTAS(CONTA\_ID)

);

CREATE TABLE EMPRESTIMO(

EMPRESTIMO\_ID INT AUTO\_INCREMENT PRIMARY KEY,

CLIENTE\_ID INT NOT NULL,

VALOR\_EMPRESTIMO DECIMAL (15,2) NOT NULL,

JUROS DECIMAL (5,2) NOT NULL,

PARCELAS INT,

DATA\_EMPRESTIMO DATE,

SITUACAO VARCHAR (15),

FOREIGN KEY (CLIENTE\_ID) REFERENCES CLIENTES(CLIENTE\_ID)

);

CREATE TABLE CARTOES(

CARTAO\_ID INT AUTO\_INCREMENT PRIMARY KEY,

CLIENTE\_ID INT,

TIPO\_CARTAO VARCHAR (10),

LIMITE DECIMAL (15,2),

NUMERO\_CARTAO CHAR (16),

DATA\_VENCIMENTO DATE NOT NULL,

FOREIGN KEY (CLIENTE\_ID) REFERENCES CLIENTES(CLIENTE\_ID)

);

CREATE TABLE INVESTIMENTOS(

INVESTIMENTO\_ID INT AUTO\_INCREMENT PRIMARY KEY,

CLIENTE\_ID INT,

TIPO\_INVESTIMENTO VARCHAR(50),

VALOR\_INVESTIDO DECIMAL(18, 2),

DATA\_INVESTIMENTO DATE,

TAXA\_RENDIMENTO DECIMAL(5, 2),

DATA\_VENCIMENTO DATE,

RENTABILIDADE\_ANUAL DECIMAL(5, 2),

FOREIGN KEY (CLIENTE\_ID) REFERENCES CLIENTES(CLIENTE\_ID)

);

CREATE TABLE FATURAS(

FATURA\_ID INT AUTO\_INCREMENT PRIMARY KEY,

CARTAO\_ID INT,

DATA\_INICIO DATE,

VENCIMENTO\_FATURA DATE,

VALOR\_FATURA DECIMAL (18,2),

JUROS\_FATURA DECIMAL (5,2),

FOREIGN KEY (CARTAO\_ID) REFERENCES CARTOES(CARTAO\_ID)

);

**Principais Adaptações para MySQL:**

* IDENTITY(1,1) foi substituído por AUTO\_INCREMENT na definição da coluna de chave primária.
* A cláusula PRIMARY KEY foi definida explicitamente junto com o AUTO\_INCREMENT.
* O comando GO utilizado no SQL Server não é necessário no MySQL.
* As restrições de chave estrangeira (FOREIGN KEY) foram definidas utilizando a sintaxe do MySQL.

**Etapa 4: Migração dos Dados (Não realizada nesta documentação)**

A migração dos dados das tabelas restantes do SQL Server para as novas tabelas no MySQL ainda precisaria ser realizada utilizando uma das seguintes abordagens (a escolha dependerá da infraestrutura e ferramentas disponíveis):

* **SQL Dump e Import:** Gerar um dump dos dados do SQL Server, adaptar o script e importar para o MySQL.
* **Ferramentas de Migração:** Utilizar ferramentas como MySQL Workbench ou outras ETLs.
* **Scripts Personalizados:** Desenvolver scripts para ler e gravar os dados entre os SGBDs.

**Próximos Passos:**

* Realizar a migração dos dados do SQL Server para o MySQL.
* Testar a integridade dos dados no novo ambiente MySQL.
* Adaptar quaisquer aplicações ou processos que utilizam o banco de dados para se conectarem e interagirem com o MySQL.
* Otimizar as consultas e o esquema no MySQL conforme necessário para garantir o desempenho adequado.

**Resumo:**

Esta documentação agora inclui tanto o modelo conceitual antigo (SQL Server) quanto o novo modelo conceitual (MySQL) do banco de dados DB\_BancoInter, ilustrando as mudanças estruturais resultantes da remoção das tabelas relacionadas a funcionários e da migração para o MySQL.