

Projeto Modelo Entidade-Relacionamento (MER)

Jayme C S Santos, Luiz Euzébio S Neto

Centro Universitário de João Pessoa – Unipê

João Pessoa, PB.

Abstract. Based on our learning and teaching to date, this content, we will initiate the construction of a model (ER) to the database of a company leasing vehicles, in order to facilitate the control of the lessor with a system management, such as internal processes rental, aiming to improve the integration with existing information that the rental company has, since the records of customers control of their vehicles and bookings. The system should register customers who wish to make leases as also record data from their vehicles, their features and general feature for both types available and the maintenance control (review) of these vehicles.

Keywords: Database, Enterprise car rental, Management System.

Resumo. Com base no nosso aprendizado e ensinamentos até o momento, neste conteúdo, vamos dar início a construção de um modelo (ER) para o Banco de Dados de uma empresa de locação de veículos, com objetivo de facilitar o controle da locadora com um sistema de gerenciamento, como os processos internos da locadora, visando melhorar a integração entre as informações existentes que a locadora possui, desde os cadastros dos clientes ao controle de seus veículos e reservas. O sistema deverá cadastrar clientes que desejam fazer locações como também registrar dados dos seus veículos, seus aspectos e característica gerais para ambos os tipos disponíveis e o controle de manutenção (revisão) destes veículos.

Palavras-chave: Banco de Dados, Empresa de locação de veículos, Sistema de Gerenciamento.

1. Introdução

1.1 História

Igualmente a muitas tecnologias na computação industrial, os fundamentos de bancos de dados relacionais surgiram na empresa IBM, nas décadas de 1960 e 1970, através de pesquisas de funções de automação de escritório. Foi durante um período da história na qual empresas descobriram que estava muito custoso empregar um número

grande de pessoas para fazer trabalhos como armazenar e indexar (organizar) arquivos. Por este motivo, valia a pena os esforços e investimentos em pesquisar um meio mais barato e ter uma solução mecânica eficiente.

Em 1970 um pesquisador da IBM - Ted Codd - publicou o primeiro artigo sobre bancos de dados relacionais. Este artigo tratava sobre o uso de cálculo e álgebra relacional para permitir que usuários não técnicos armazenassem e recuperassem grande quantidade de informações. Codd visionava um sistema onde o usuário seria capaz de acessar as informações através de comandos em inglês, onde as informações estariam armazenadas em tabelas.

Devido à natureza técnica deste artigo e a relativa complicação matemática, o significado e proposição do artigo não foram prontamente realizados. Entretanto, ele levou a IBM a montar um grupo de pesquisa conhecido como System R (Sistema R).

O projeto do Sistema R era criar um sistema de banco de dados relacional o qual eventualmente se tornaria um produto. Eventualmente o Sistema R evoluiu para SQL/DS, o qual posteriormente tornou-se o DB2. A linguagem criada pelo grupo do Sistema R foi a Structured Query Language (SQL) - Linguagem de Consulta Estruturada). Esta linguagem tornou-se um padrão na indústria para bancos de dados relacionais e hoje em dia é um padrão ISO (International Organization for Standardization). A ISSO é a Organização Internacional de Padronização.

1.2 Conceitos Fundamentais de Banco de Dados

Um banco de dados “É uma coleção de dados inter-relacionados, representando informações sobre um domínio específico” (Korth, H.F. e Silberschatz, 1994), ou seja, sempre que for possível agrupar informações que se relacionam e tratam de um mesmo assunto, posso dizer que tenho um banco de dados.

Podemos exemplificar situações clássicas como uma lista telefônica, um catálogo de CDs ou um sistema de controle de RH de uma empresa.

Já um sistema de gerenciamento de banco de dados (SGBD) é um software que possui recursos capazes de manipular as informações do banco de dados e interagir com o usuário. Exemplos de SGBDs são: Oracle, SQL Server, DB2, PostgreSQL, MySQL, o próprio Access ou Paradox, entre outros.

Podemos conceituar um sistema de banco de dados como o conjunto de quatro componentes básicos: dados, hardware, software e usuários. “Sistema de bancos de dados pode ser considerado como uma sala de arquivos eletrônica”. (Date, C.J., 1991).

Os objetivos de um sistema de banco de dados são o de isolar o usuário dos detalhes internos do banco de dados (promover a abstração de dados) e promover a independência dos dados em relação às aplicações, ou seja, tornar independente da aplicação, a estratégia de acesso e a forma de armazenamento.

1.3 Abstração de dados

O sistema de banco de dados deve garantir uma visão totalmente abstrata do banco de dados para o usuário, ou seja, para o usuário do banco de dados pouco importa qual unidade de armazenamento está sendo usada para guardar seus dados, contanto que os mesmos estejam disponíveis no momento necessário.

Esta abstração se dá em três níveis:

- Nível de visão do usuário: as partes do banco de dados que o usuário tem acesso de acordo com a necessidade individual de cada usuário ou grupo de usuários;
- Nível conceitual: define quais os dados que estão armazenados e qual o relacionamento entre eles;
- Nível físico: é o nível mais baixo de abstração, em que define efetivamente de que maneira os dados estão armazenados.

1.4 Projeto de banco de dados

Todo bom sistema de banco de dados deve apresentar um projeto, que visa a organização das informações e utilização de técnicas para que o futuro sistema obtenha boa performance e também facilite infinitamente as manutenções que venham a acontecer.

O projeto de banco de dados se dá em duas fases:

- Modelagem conceitual;
- Projeto lógico.

1.4.1 Modelo conceitual

É a descrição do BD de maneira independente ao SGBD, ou seja, define quais os dados que aparecerão no BD, mas sem se importar com a implementação que se dará ao BD. Desta forma, há uma abstração em nível de SGBD.

1.4.2 Modelo lógico

Descreve o BD no nível do SGBD, ou seja, depende do tipo particular de SGBD que será usado. Não podemos confundir com o Software que será usado. O tipo de SGBD que o modelo lógico trata é se o mesmo é relacional, orientado a objetos, hierárquico, etc.

1.5 Modelo Entidade Relacionamento (MER)

O Modelo Entidade-Relacionamento tem o objetivo de representar as estruturas de dados da forma mais próxima do mundo real dos negócios.

Existem três conceitos no Modelo Entidade Relacionamento: Entidade, Atributo e Relacionamento.

Entende-se que **Entidade** são objetos, as características dos objetos são os **Atributos**, e a relação entre os objetos são os **relacionamentos**.

1.5.1 Entidade

Uma entidade serve tanto para depósito quanto para recuperação de dados. Ela representa substantivos, concretos ou abstratos. Exemplo: Na figura 1, todas as pessoas físicas de um determinado cadastro de uma empresa podem ser armazenadas em uma entidade chamada Pessoa Física.

Todas as pessoas da Entidade Pessoa Física podem ter um identificador único.

Assim como a Entidade é conhecida por Tabela, o identificador seria a chave primária.

Exemplo:

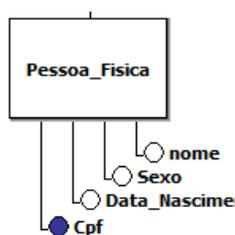


Figura 1: exemplo de entidade

1.5.2 Atributo

Toda entidade possui atributos, ou seja, características que informam sobre a entidade.

A entidade Pessoa Física contém informações sobre Nome, Sexo, Data de Nascimento de cada pessoa, identificado entre os outros por uma chave primária, no caso pode ser um atributo Cpf. (Ver figura 1)

Tipos de atributos

Atributo Simples

Não possui qualquer característica especial. A maioria dos atributos serão simples. Quando um atributo não é composto, recebe um valor único como nome, por exemplo e não é um atributo chave, então ele será atributo simples.

Atributo Composto

O seu conteúdo é formado por vários itens menores. Exemplo: Endereço. Seu conteúdo poderá ser dividido em vários outros atributos, como: Rua, Número, Complemento, Bairro, Cep e Cidade. Este tipo de atributo é chamado de atributo composto. Veja o exemplo na figura 2.

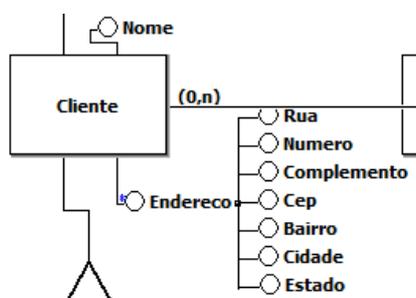


Figura 2 – Endereco: exemplo de atributo composto

Atributo Multivalorado

O seu conteúdo é formado por mais de um valor.

Exemplo: Telefone, como mostra a figura 3. Uma pessoa poderá ter mais de um número de telefone. O atributo multivalorado serão tratados com mais detalhes na normalização de dados.

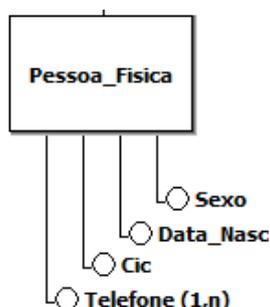


Figura 3 – Telefone: exemplo de atributo multivalorado

Atributo Determinante (chave)

Identifica de forma única uma entidade, ou seja, não pode haver dados repetidos.

É indicado sublinhando-se o nome do atributo, no MER, uma bolinha fechada. Exemplo: CNPJ, CPF, Código do cliente, Número da matrícula, etc. Os atributos determinantes serão as chaves primárias no banco de dados e seu uso tem implicações na normalização de dados, como mostra a figura 4.

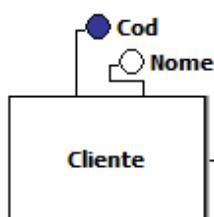


Figura 4 – Cod: exemplo de atributo chave

1.6 Modelo de Dados Relacional

Introduzido por Codd (1970) é, dentre os modelos de dados de implementação, o modelo relacional mais simples, com estrutura de dados uniforme, e também o mais formal.

Um banco de dados é um aplicativo que pode armazenar e recuperar dados muito rapidamente. O bit relacional refere-se à forma como os dados são armazenados no banco de dados e como ele está organizado. Quando falamos sobre o banco de dados, falamos sobre um banco de dados relacional, (na verdade um SGBDR - Sistema Gerenciador de Banco de Dados Relacionais).

A principal construção para representação dos dados é a relação, uma tabela com linhas não ordenadas e colunas. Uma relação consiste de um esquema e de uma instância. O esquema especifica o nome da relação e o nome e o domínio de cada coluna, também denominada atributo ou campo da relação. O domínio do atributo é referenciado no esquema por seu nome e serve para restringir os valores que este atributo pode assumir. O esquema de uma relação é invariável ao longo do tempo, sendo modificado apenas por comandos específicos. Um exemplo de esquema de relação é:

Cliente (Cod: int, Nome: string, Endereco: string)

Neste caso está sendo definida a relação de nome Cliente, com atributos Cod, Nome, e Endereco, cujos domínios são respectivamente int, string e string.

A instância de uma relação é o conjunto de linhas, também denominadas tuplas ou registros, distintas entre si, que compõem a relação em um dado momento. Ela é

variável, já que o número de tuplas e o conteúdo de seus atributos podem variar ao longo do tempo. A instância de uma relação deve seguir sempre o seu respectivo esquema, respeitando o número de atributos definidos, bem como os seus domínios. Esta restrição, denominada restrição de domínio, é muito importante. O modelo relacional somente considera relações que satisfaçam esta restrição. Um exemplo de uma instância para o esquema Cliente é ilustrado na Figura 5.

Cod	Nome	Endereco
122011	João	Rua Montreal
148012	Maria	Rua Araruna
184450	José	Rua Castro Alves

Figura 5 – Exemplo de instância da relação Cliente

2. Desenvolvimento

Como já mencionado, a seguir veremos a construção de um modelo (ER) para o Banco de Dados de uma empresa de locação de veículos, onde deverá cadastrar clientes, registrar dados dos seus veículos, dentre outras necessidades.

A locadora possui de clientes, de automóveis, reservas, filiais e etc. Para organizar o sistema da locadora construiremos um MER, assim podendo ser organizado e armazenado no BD as informações necessárias. Composto o MER temos: entidades, relacionamentos, agregação, especialização/generalização, cardinalidades, e atributos como pode ser visto o modelo a seguir na figura 6:

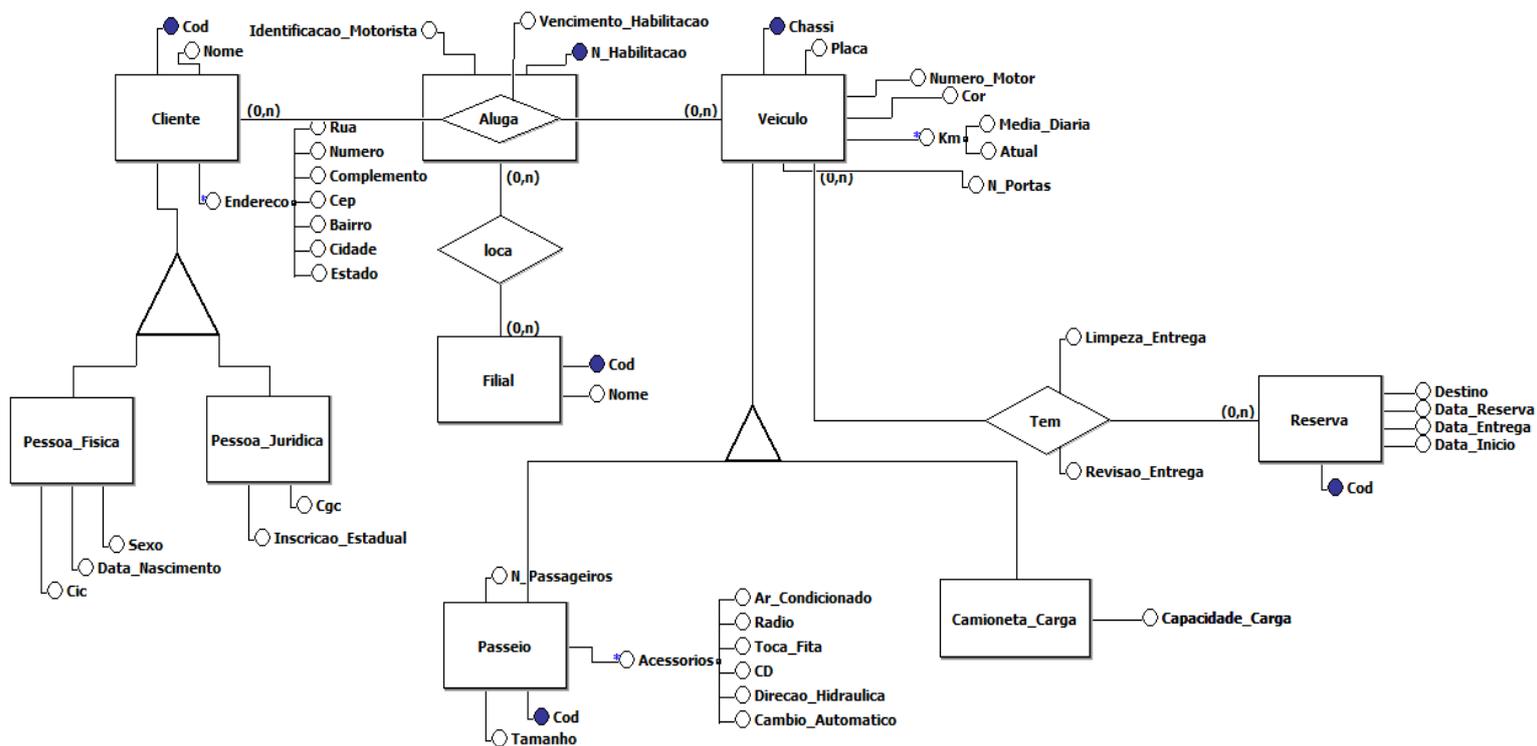


Figura 6 – Modelo (ER) para o Banco de Dados de uma empresa de locação de veículos

De acordo com o Modelo (ER) apresentado acima:

2.1 Os Atributos

Dentre os tipos de atributos vistos neste artigo, sendo eles as propriedades que descrevem entidades, é visível no modelo acima inúmeros atributos que descrevem, identificam e caracterizam as respectivas entidades. Observando o relacionamento TEM entre as entidades VEÍCULO e RESERVA, Há também os atributos de relacionamento que, depende de todos os conjuntos - entidade associados entre si.

2.2 As Entidades

Na entidade Cliente (Figura 7) podemos observar que temos atributos chave, simples, e composto. O atributo (**Cod**), é um atributo chave. O atributo (**Nome**), um atributo simples. O atributo (**Endereco**), é um atributo composto, assim ele possui mais atributos simples que são: Rua, Numero, Complemento, Cep, Bairro, Cidade, e Estado. Essa entidade cliente possui uma especialização/generalização, cliente vai fazer uma herança de atributos para as outras entidades que está nessa especialização.

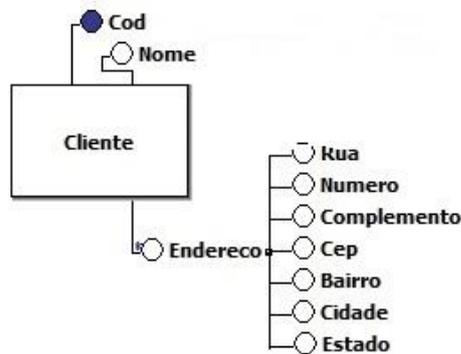


Figura 7 – Entidade Cliente com seus atributos.

Entidade Pessoa_Fisica (Figura 8), ela é um entidade generalizada de Cliente, assim, herda os atributos de Cliente. Observando os atributos, um chave (**Cic**), dois simples (**Sexo**) e (**Data_Nascimento**).

Entidade Pessoa_Juridica (Figura 8), é do mesmo tipo da anterior, uma entidade generalizada de Cliente. Observando os atributos temos, um chave (Cgc), e um simples (Inscricao_Estadual).

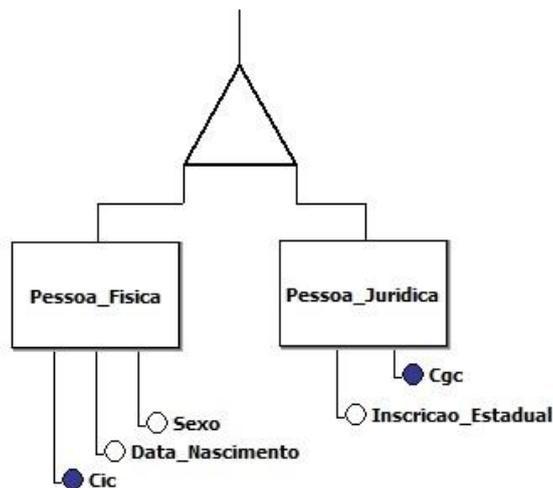


Figura 8 – Especialização/Agregação da entidade cliente.

Na entidade Filial (Figura 9), essa entidade pertence a uma agregação com as entidades Cliente e Veiculo. Possui dois atributos um chave (**Cod**) e um simples (**Nome**).

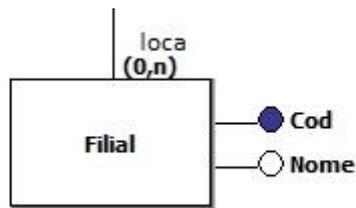


Figura 9 – Entidade filial com seus atributos.

Na entidade Veiculo (Figura 10), possui vários atributos sendo, o chave (**Chassi**), os simples (**Numero_Motor**), (**Cor**), (**N_Portas**), e um composto (**Km**) que possui mais dois atributos simples (**Media_Diaria**), e (**Atual**).

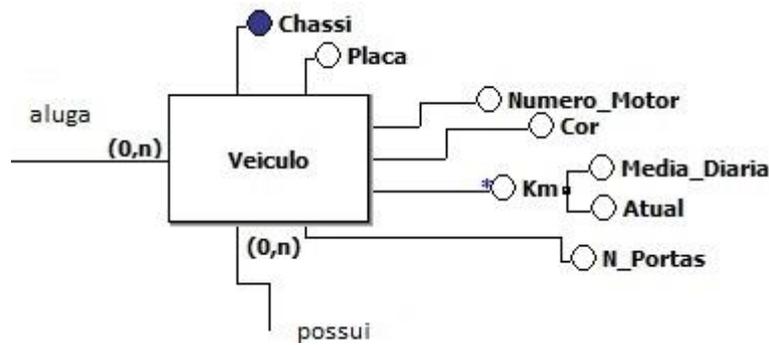


Figura 10 – Entidade veiculo com seus atributos.

Na entidade Tipo_Veiculo (Figura 11), ela tem um relacionamento com outra entidade e tem uma generalização.

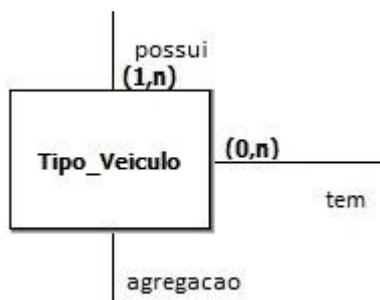


Figura 11 – Entidade tipo veiculo.

A entidade Passeio (Figura 12), é uma generalização de Tipo_Veiculo, ela também possui uma generalização, os atributo dessa entidade são: um simples (**N_Passageiros**), e um composto (**Acessorios**) que tem os seguintes atributos simples (**Ar_Condicionado**), (**Radio**), (**Toca_Fita**), (**CD**), (**Direcao_Hidraulica**), e (**Cambio_Automatico**).

A entidade Camioneta_Carga (Figura 12), é uma generalização de Tipo_Veiculo, que possui apenas uma entidade simples (**Capacidade_Carga**).

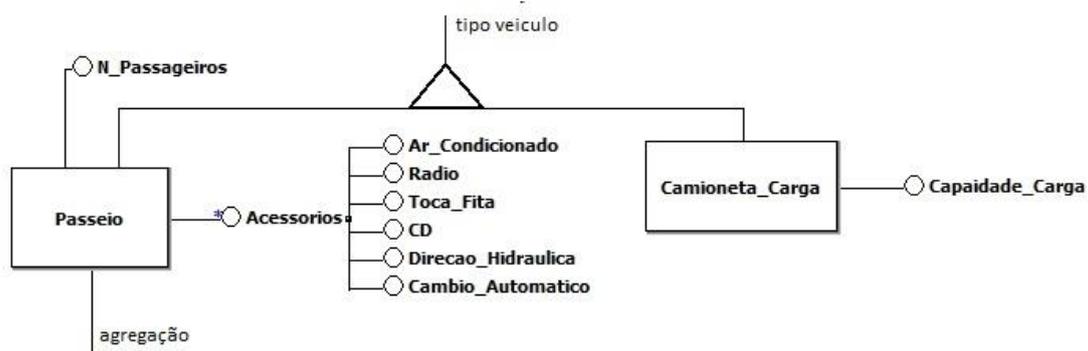


Figura 12 – Agregação da entidade tipo veículo.

As generalizações da entidade Passeio são dois, Automovel e Camioneta_Passeio (Figura 13). A entidade Automovel possui um atributo simples (Tamanho), e a entidade Camioneta_Passeio possui um atributo simples (Capacidade_Carga).

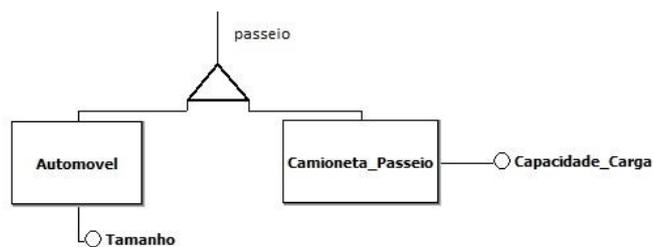


Figura 13 – Agregação da entidade passeio.

A entidade Reserva (Figura 14), possui como atributo chave (Cod), e simples (Destino), (Data_Reserva), (Data_Entrega) e (Data_Inicio).



Figura 14 – Entidade reserva.

Os Relacionamentos

Vale ressaltar que, relacionamento é uma associação entre entidades, cada tipo entidade que participa de um tipo relacionamento executa um papel no relacionamento.

No relacionamento Aluga (Figura 15), esse relacionamento é uma agregação¹ que agrega a entidade Cliente, Filial e Veiculo. Essa agregação faz o relacionamento com a entidade Filial com o relacionamento Loca. O Aluga relacionamento possui seus

próprios atributos um chave (N_Habilitacao), e os simples (Vencimento_Habilitacao) e (Identificacao_Motorista).

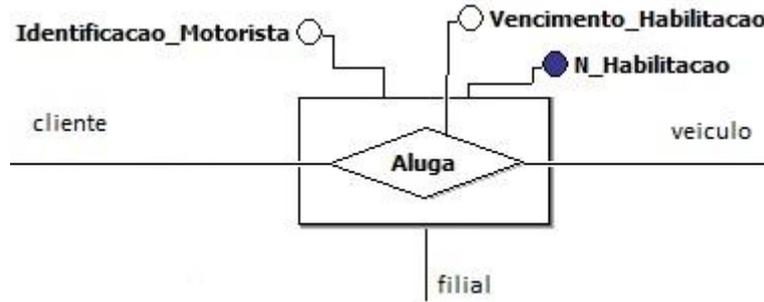


Figura 15 – Agregação que liga as entidades cliente, filial e veiculo.

No relacionamento Possui (Figura 16), esse relacionamento faz a relação do relacionamento Veiculo com o relacionamento Tipo_Veiculo.

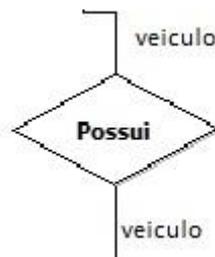


Figura 16 – O Relacionamento Possui.

O relacionamento Tem (Figura 17), faz a relação da entidade Tipo_Veiculo com a entidade Reserva.

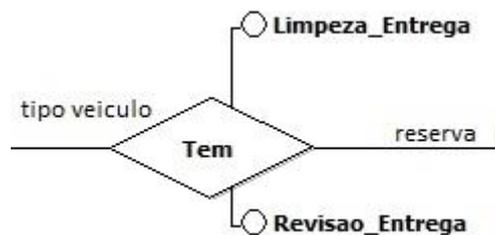


Figura 17 – O Relacionamento Tem com os seus atributos.

1agregação “agrega duas ou mais entidades no conceito de entidade”

Visto os conceitos básicos e práticas de construção de um modelo relacional, segue abaixo, o Modelo Relacional deste projeto para o Banco de Dados de uma empresa de locação de veículos.

Cliente (Cod, Nome, Endereco)

Cliente_Endereco (Cod, Rua, Numero, Complemento, Cep, Bairro, Cidade, Estado)

Cod referencia Cliente

Pessoa_Fisica (Cod, Cic, Data_Nascimento, Sexo)

Cod referencia Cliente

Pessoa_Juridica (Cod, Cgc, Inscricao_Estadual)

Cod referencia Cliente

Veiculo (Chassi, placa, Numero_Motor, Cor, N_Portas, Media_Diaria, Atual)

Cliente_Veiculo (cliente, veiculo, N_Habilitacao, Vencimento_Habilitacao, Identificacao_Motorista)

cliente referencia Cliente

veiculo referencia Veiculo

Filial (Cod, Nome)

Cliente_Filial (cliente, filial)

cliente referencia Cliente

filial referencia Filial

Passeio (Chassi, Cod, Tamanho, N_Passeio, Acessorios)

Chassi referencia Veiculo

Passeio_Acessorios (Chassi, Cod, Ar_Condicionado, Radio, Toca_Fita, CD, Direcao_Hidraulica, Cambio_Automatico)

Chassi referencia Veiculo

Cod referencia Passeio

Reserva (Cod, Destina, Data_Reserva, Data_Entrega, Data_Inicio)

Veiculo_Reserva (veiculo, reserva, Limpeza_Entrega, Revisao_Entrega)

veiculo referencia Veiculo

reserva referencia Reserva

Considerações Iniciais

Com base neste modelo (ER) apresentado, foram abordados os principais comandos de um banco de dados. Assim, procurou atender aos objetivos, permitindo a facilidade no acesso e organização dos dados, atendendo aos requisitos na elaboração do sistema de banco de dados da empresa de locação de veículos.

Nesta segunda etapa do projeto foi agregado o conceito de modelo de dados relacional, que se baseia no princípio em que todos os dados sejam guardados em tabelas (relações), garantindo assim, acesso facilitado aos dados, tornando possível ao usuário, a utilização de uma grande variedade de práticas no tratamento das informações.

Referências Utilizadas

Date, C.J.; Int. a Sistemas de Bancos de Dados, tradução da 4a.edição norte-americana, Editora Campus, 1991.

C. J. Date. Introdução à Sistema de Banco de Dados. Editora Campus, 2000.

Korth, H.F. e Silberschatz, A.; Sistemas de Bancos de Dados, Makron Books, 2a. edição revisada, 1994.

Guimarães, C. C. (2003) “Fundamentos de banco de dados: modelagem, projeto e linguagem SQL”, Editora da Unicamp.