



Universidade do Minho

Mestrado Integrado em Engenharia Informática
Licenciatura em Ciências da Computação

Unidade Curricular de Bases de Dados

Ano Lectivo de 2016/2017

Pedro Dias Imobiliária

João Dias, Luís Silva, Nelson Sá, Tiago Loureiro

Novembro, 2016

BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

Pedro Dias Imobiliária

**João Dias a 66850, Luís Silva a72113,
Nelson Sá a70925, Tiago Loureiro a71191**

Novembro, 2016

ÍNDICE

RESUMO	V
ÍNDICE DE FIGURAS	VI
ÍNDICE DE TABELAS	VII
INTRODUÇÃO	1
CONTEXTUALIZAÇÃO.....	1
APRESENTAÇÃO DO CASO DE ESTUDO	1
MOTIVAÇÃO E OBJECTIVOS.....	2
ESTRUTURA DO RELATÓRIO.....	2
ANÁLISE DE REQUISITOS	3
REQUISITOS DE DADOS E TRANSAÇÕES	3
PERGUNTAS RELEVANTES	4
TRANSAÇÕES RELEVANTES	4
MODELAÇÃO CONCEPTUAL DA BASE DE DADOS.....	5
DESENHO DO DIAGRAMA	5
LISTAGEM DAS ENTIDADES EXISTENTES.....	6
RELACIONAMENTOS EXISTENTES	6
IDENTIFICAÇÃO DE ATRIBUTOS E SEUS DOMÍNIOS	6
DETERMINAÇÃO DAS CHAVES PRIMÁRIAS E CANDIDATAS.....	9
VERIFICAÇÃO E VALIDAÇÃO DO MODELO DESENVOLVIDO COM AS TRANSAÇÕES DOS UTILIZADORES	10
REVISÃO E VALIDAÇÃO DO MODELO COM O CLIENTE	10
MODELAÇÃO LÓGICA DE BASE DE DADOS.....	12
ESQUEMA LÓGICO	12
DERIVAÇÃO DE RELACIONAMENTOS PARA O MODELO DE DADOS LÓGICO	12
<i>Tipos de Entidades Fortes.....</i>	<i>13</i>
<i>Tipos de entidades fracas.....</i>	<i>13</i>
<i>Relacionamentos binários 1:N.....</i>	<i>13</i>
<i>Relacionamentos binários 1:1</i>	<i>13</i>
<i>Remoção de Relacionamentos N:M.....</i>	<i>14</i>
<i>Remoção de Relacionamentos Complexos.....</i>	<i>14</i>

<i>Atributos Multivalor</i>	14
<i>Atributos compostos</i>	14
VALIDAÇÃO DE RELAÇÕES ATRAVÉS DA NORMALIZAÇÃO	14
VALIDAÇÃO DE RELAÇÕES CONTRA AS TRANSAÇÕES DOS UTILIZADORES	15
VERIFICAÇÃO DE RESTRIÇÕES DE INTEGRIDADE	16
<i>Restrição de Dados</i>	16
<i>Restrições de domínio de atributos</i>	16
<i>Multiplicidade</i>	16
<i>Integridade de entidades</i>	16
<i>Integridade referencial</i>	16
CONSIDERAÇÕES SOBRE O CRESCIMENTO FUTURO	17
MODELAÇÃO FÍSICA DA BASE DE DADOS	18
SISTEMA DE GESTÃO DE BASE DE DADOS	18
ESTRUTURAS DAS RELAÇÕES	18
RESTRIÇÕES GERAIS	22
ORGANIZAÇÃO DE FICHEIROS E ÍNDICES	25
<i>Análise de Transações</i>	25
RESPOSTA ÀS PERGUNTAS DE VALIDAÇÃO FEITAS PELO UTILIZADOR	26
ESCOLHA E CRIAÇÃO DE ÍNDICES	28
ANÁLISE DE ESPAÇO EM DISCO	28
MECANISMOS DE SEGURANÇA	29
FERRAMENTAS UTILIZADAS	30
CONCLUSÃO	31
REFERÊNCIAS	32
LISTA DE SIGLAS E ACRÓNIMOS	33

Resumo

Neste trabalho, nós começamos por identificar o problema e o contexto em que se enquadra, a seguir determinamos os requisitos para serem analisados, para que com os mesmos sejam alcançados os objetivos principais. Numa outra fase, prosseguimos com a implementação do modelo conceptual deste trabalho, com base nas informações recolhidas anteriormente.

Depois, fizemos a transição do modelo conceptual para o modelo lógico, aperfeiçoando a estrutura inicial, até esta ficar solidificada. O modelo foi validado conforme o que os diferentes utilizadores poderão manipular na BD.

Na ultima fase do trabalho, que foi a fase final do projeto, procedemos à implementação física da base de dados.

Área de Aplicação: Desenho e arquitetura de uma Base de Dados para uma imobiliária.

Palavras-Chave: Imobiliária, Clientes, Funcionarios, Imóveis, Moradia, Apartamento, Loja, Terreno, Transações, Base de Dados, Entidades, Atributos, Relacionamentos.

Índice de Figuras

FIGURA 1. MODELO CONCEPTUAL.....	5
FIGURA 2. MODELO LÓGICO	12

Índice de Tabelas

TABELA 1 - PERMISSÕES	3
TABELA 2 - ENTIDADES	6
TABELA 3 - RELACIONAMENTOS	6
TABELA 4 - ATRIBUTOS DE FUNCIONARIO	7
TABELA 5 - ATRIBUTOS DE CLIENTE	8
TABELA 6 - ATRIBUTOS DE IMOVEL	9
TABELA 7 - ATRIBUTOS DE TRANSACAO	9
TABELA 8 - CHAVES PRIMÁRIAS E CANDIDATAS	10

1 Introdução

Neste capítulo é dada uma breve apresentação do trabalho, o contexto do problema a resolver, e os motivos para realização do projeto e os objetivos que nos propomos a alcançar com o mesmo.

O projeto que nos foi proposto pelo empresário Pedro Dias, tem o intuito de agilizar e melhorar o armazenamento e acesso da informação relativa a funcionários, clientes, imóveis e transações.

1.1 Contextualização

Uma empresa imobiliária fundada pelo Sr. Pedro Dias, chamada "Pedro Dias Imobiliária".

Empresa com 2 anos de atividade, que só agora tem capacidade monetária para a criação de uma base de dados informática.

Pedro Dias, como já foi referido dono da imobiliária "Pedro Dias Imobiliária", vem pensando na criação de uma base de dados para a sua imobiliária. A ideia surgiu porque ele estava cansado de acumular capas com a papelada de imóveis vendidos, alugados e de informação sobre transações e clientes. Depois de trocar algumas impressões com os seus sócios, Pedro Dias verificou que já era comum em muitas imobiliárias terem uma base de dados, então ele e os seus parceiros meteram as mãos a obra para criação da sua base de dados para agilizar processos e modernizar a sua Imobiliária. A sua criação tinha de ser feita com rapidez caso quisesse ficar em pé de igualdade com os seus concorrentes ou então via a perda de clientes como realidade.

1.2 Apresentação do Caso de Estudo

Uma base de dados é, em termos gerais uma coleção de dados partilhados, que estão logicamente relacionados e organizados de acordo com os requisitos de informação de uma organização, ou seja, permite uma melhor organização detalhada de informação, com o objetivo de permitir a manipulação e consulta da mesma pelos gestores e utilizadores. Ao mesmo tempo que seja robusta e de fácil acesso e procura. Aplicaremos então uma base de dados ao caso em estudo, uma Imobiliária.

A organização desta Imobiliária deverá ser estruturada de forma a que um Cliente (identificação, nome, contribuinte, data de nascimento, contacto(telemóvel,email)) quisesse efetuar uma compra/aluguer, só tinha que se dirigir a imobiliária em questão falar com o Funcionário (identificação, nome, data de nascimento, contacto(telemóvel, email)) que lhe disponibilizaria a carteira de imóveis, dependendo do que queria no leque de hipóteses que a imobiliária oferece, moradias (identificação, morada, valor de venda, área),

apartamentos(identificação, morada, área, tipologia, valor de venda), loja (id, morada, área, valor venda) e terreno (id, morada, área, valor venda), após requerer o que deseja, haverá uma transação (identificação da Transação, tipo, se é um aluguer, compra ou venda, identificação do Imóvel, data inicial, data final, identificação do Cliente, identificação do Funcionário, identificação do Funcionário, valor) entre Cliente e Funcionário em questão, que continha o tipo de imóvel comprado/alugado, data em que foi transacionado e valor monetário envolvido na transação.

1.3 Motivação e Objectivos

Apesar de a imobiliária funcionar em formato papel, a possibilidade verificada pelo dono, Pedro Dias, de que podia ser representada em formato digital, possibilitaria ao mesmo uma manipulação eficaz e rápida da informação sobre clientes, imóveis e transações.

A melhor forma seria então construir uma base de dados para a imobiliária em questão, alcançando como objetivos, a rapidez de acesso à informação, anteriormente falada, por parte dos Funcionários, que haja uma diferenciação de utilizadores entre Funcionário/gestor e Cliente/utilizador, aplicando assim a um caso real de quotidiano de uma imobiliária de forma a que nos permita ver o comportamento que traz à organização dos dados, para que no fim consigamos gerir e construir uma base de dados funcional e devidamente organizada.

1.4 Estrutura do Relatório

Neste ponto do relatório, faremos uma breve apresentação daquilo que iremos incluir futuramente. Ao longo do documento em causa iremos encontrar toda a informação necessária à construção de uma base de dados de uma Imobiliária, bem como a sua estrutura. Iremos encontrar os modelos conceptual contruídos para a nossa base de dados, bem como as entidades em causa e os relacionamentos estabelecidos entre elas.

2 Análise de Requisitos

Antes de proceder à construção da base de dados, é necessário compreender como deverá funcionar uma imobiliária e quais são os pedidos do cliente.

Ora, uma imobiliária é uma empresa que atua no mercado imobiliário, intermediando a venda ou aluguer de diversos tipos imoveis. Pela venda ou aluguer, a imobiliária ganha uma comissão aplicada ao valor total pago pela pessoa que dispõe o imóvel para venda, vulgo vendedor.

O ato de disponibilização de um imóvel para venda ou aluguer é feito pelo vendedor e intermediado por um funcionário. Sendo que o ato de compra ou aluguer parte do comprador sendo, mais uma vez, intermediado por um funcionário.

Terá de haver também espaço para o registo de todas as transações, que discriminam o tipo, o comprador, o vendedor e o intermediário.

2.1 Requisitos de Dados e Transações

Como já foi dito, esta Imobiliária deverá ser estruturada de forma a que um cliente que tenha interesse em comprar/vender/alugar um imóvel, apenas tenha que se dirigir à imobiliária e falar com o funcionário. Logo, os funcionários serão os únicos utilizadores desta base de dados.

ENTIDADE	ACESSO	
FUNCIONÁRIO	Adicionar	NÃO
	Alterar	NÃO
	Remover	NÃO
	Visualizar	SIM
Cliente	Adicionar	SIM
	Alterar	SIM
	Remover	NÃO
	Visualizar	SIM
Imovel	Adicionar	SIM
	Alterar	SIM
	Remover	NÃO
	Visualizar	SIM
Transacao	Adicionar	SIM
	Alterar	SIM
	Remover	NÃO
	Visualizar	SIM

Tabela 1 - Permissões

Depois de um levantamento rigoroso daquilo que é pretendido que seja feito pela base de dados, conclui-se que esta deverá respeitar a seguinte lista de requisitos:

- Cada funcionário deverá estar registado na base de dados com um nome e um contacto (telemóvel e/ou email).
- Cada cliente deverá ser registado com um nome, numero de contribuinte e contacto (telemóvel e/ou email).
- Cada imóvel deverá ter a informação da sua dimensão, valor, morada, proprietário e se é para vender ou alugar.
- Cada transação deverá ser registada com informação sobre o imóvel, valor, data e se é uma compra ou aluguer.
- Uma transação de aluguer deverá ter, além de uma data inicial, uma data final. A informação do valor será do valor total pago ao longo do período de aluguer.
- Uma transação de venda deverá só ter a data inicial que corresponde a assinatura do contrato.
- Um aluguer só se pode fazer com um mínimo de um ano de contrato.
- Uma transação terá de ter associados um cliente comprador e um funcionário.

2.2 Perguntas Relevantes

Depois do levantamento de requisitos e respetiva análise, foram elaboradas várias perguntas de forma a garantir que o sistema responde às necessidades da imobiliária.

- 1) É possível adicionar imóveis à base de dados?
- 2) É possível criar (adicionar) ficha cliente?
- 3) É possível adicionar um novo funcionário?
- 4) Qual funcionário foi/está responsável por determinada transação?
- 5) Quantos imóveis estão à venda?
- 6) Quais imóveis estão disponíveis para aluguer?
- 7) Quais imóveis estão vendidos?
- 8) Quais imóveis estão alugados?
- 9) Qual o valor de determinado imóvel?
- 10) Quais imóveis pertencentes a uma certa cidade?

2.3 Transações Relevantes

Como foi dito no capítulo 2.2, os funcionários da Imobiliária serão os únicos utilizadores do sistema e terá de ser possível, a estes, manipular e consultar todos os dados relacionados com clientes, imóveis e transações.

- Inserir um novo Cliente;
- Inserir um novo Funcionário;
- Inserir um nova Transação;
- Introduzir imóveis na BD;
- Comprar, alugar, por à venda e por à alugar imóveis;
- Verificar o numero de vendas num mês.

3 Modelação Conceptual da Base de Dados

Neste capítulo passamos a enunciar o esquema essencial para modelação de uma base de dados, que é modelo conceptual.

O modelo conceptual produz o esquema conceptual, expresso numa linguagem independente da base de dados que irá ser usada. Este esquema utiliza noções abstratas que permitem capturar os requisitos do problema, sem tornar decisões dependentes quaisquer sistemas ou tecnologias. Nesta fase do projeto, utilizam-se modelos ditos semânticos, que tentam capturar o significado dos dados que serão necessários. Por ser um modelo semântico, é fácil envolver utilizadores com perfis de conhecimento diferentes, uma vez que só estamos preocupados com a forma como os utilizadores se apercebem dos dados e os querem utilizar. Um dos modelos mais utilizados nesta fase é o modelo Entidade-Relacionamento (modelo E-R) [1].

Vamos então apresentar o esquema conceptual descrevendo as entidades, atributos e relacionamentos e a seguir identificamos as chaves candidatas a primárias e estrangeiras.

3.1 Desenho do Diagrama

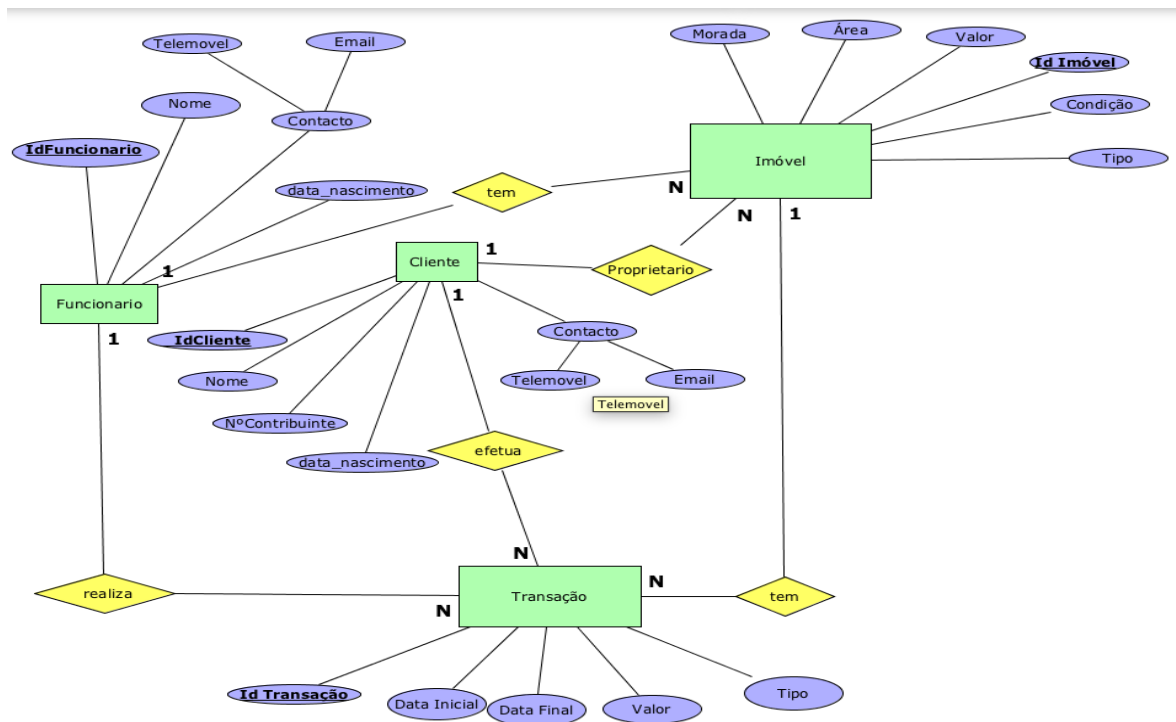


Figura 1. Modelo Conceptual

3.2 Listagem das Entidades Existentes

Neste trabalho identificamos 4 entidades, sendo elas as seguintes:

- Cliente
- Funcionário
- Imóvel
- Transação

Entidade	Descrição
Funcionário	Descreve a informação sobre cada Funcionário da imobiliária.
Cliente	Descreve a informação sobre cada Cliente.
Imóvel	Descreve a informação sobre cada imóvel da imobiliária.
Transação	Descreve a informação sobre cada transação efetuada na imobiliária.

Tabela 2 - Entidades

3.3 Relacionamentos Existentes

A informação sobre os relacionamentos criados no nosso modelo conceptual.

Entidade 1	Nome do Relacionamento	Entidade 2	Cardinalidade
Funcionário	tem	Imóvel	1:N
Funcionário	realiza	Transação	1:N
Cliente	efetua	Transação	1:N
Imóvel	tem	Transação	1:N
Cliente	proprietário	Imóvel	1:N

Tabela 3 - Relacionamentos

3.4 Identificação de atributos e seus domínios

As tabelas seguintes contêm informações relativas aos atributos de cada entidade. Em cada tabela é referido o nome do atributo, uma breve descrição do atributo, seguida do tipo de dados, do seu domínio de valores e ainda se o atributo pode ou não ser nulo.

Funcionário:

Atributo	Descrição	Tipo de Dados	Domínio de valores	NULL
idFuncionario	Código de identificação do funcionario	Número	Número sequencial positivo	Não
Contacto	Atributo composto que armazena informações de contacto do Funcionario	Atributo composto	Constituído por um email sequência de caracteres e de um numero de telemóvel, sequencia de números	Não
Nome	Nome do Funcionario	Sequencia de caracteres	Sequência de caracteres (tamanho máximo 45)	
Data_Nasc	Data de nascimento	Data	Numero positivo com uma formatação do tipo "aaaammdd"	Não

Tabela 4 - Atributos de Funcionario

Cliente:

Atributo	Descrição	Tipo de Dados	Domínio de Valores	NULL
idCliente	Código de identificação do Cliente	Número	Número sequencial positivo	Não
Contacto	Atributo composto e que armazena informações de contacto	Atributo composto	Constituído por um email sequência de caracteres e de um numero de telemóvel,	Não

	do Cliente		sequencia de números	
Nome	Nome do Cliente	Sequencia de caracteres	Sequência de caracteres (tamanho máximo 45)	Não
Contribuinte	Número de Identificação Fiscal do Cliente	Número	Número sequencial positivo	Não
Data_Nasc	Data de nascimento	Data	Numero positivo com uma formatação do tipo “aaaammdd”	Não

Tabela 5 - Atributos de Cliente

Imóvel:

Atributo	Descrição	Tipo de Dados	Domínio de Valores	NULL
idImóvel	Código de identificação do imóvel	Número	Numero sequencial positivo	Não
Área	Área em m2 do imóvel	Numero	Numero sequencial positivo	Não
Morada	Morada do imóvel	Sequência de caracteres	Sequência de caracteres (tamanho máximo 50)	Não
Tipo	Tipo de imóvel, Moradia, Apartamento, Loja e Terreno	Sequência de caracteres	Sequência de caracteres do tipo(Moradia, Apartamento, Loja, Terreno) (tamanho máximo 12)	Não
Condição	Condição do imóvel, se está alugado, vendido ou por vender	Sequência de caracteres	Sequência de caracteres do tipo (Aluguer, Alugado, Vendido,	Não

			Venda) (tamanho máximo 8)	
Valor	Valor monetário do imóvel	Número	Numero inteiro positivo	Não

Tabela 6 - Atributos de Imovel

Transação:

Atributo	Descrição	Tipo de Dados	Domínio de Valores	NULL
idTransação	Numero de identificação da transação	Número	Número sequencial positivo	Não
Tipo	Se é uma venda ou um aluguer	Sequência de caracteres	Sequência de caracteres do tipo aluguer ou venda) (tamanho máximo 45)	Não
Data_Inicial	Data inicial do aluguer, ou data única na venda	Data	Numero positivo com uma formatação do tipo “aaaammdd”	Não
Valor	Valor monetário da transação	Número	Número inteiro positivo	Não
Data_Final	Data final do aluguer	Data	Numero positivo com uma formatação do tipo “aaaammdd”	Sim

Tabela 7 - Atributos de Transacao

3.5 Determinação das chaves primárias e candidatas

Todas as entidades referenciadas anteriormente necessitam de uma chave primária, de modo a se conseguir se identificar exclusivamente com um único atributo.

Na próxima tabela, iremos identificar as chaves primárias que decidimos utilizar, explicar o motivo para qual a escolhermos, caso existam, as restantes chaves candidatas.

Entidade	Chave primária	Motivo	Chaves alternativas
Cliente	idCliente	Cada Cliente tem um número de identificação único	Contribuinte
Funcionário	idFuncionario	Cada Funcionário tem um número de identificação único	Contacto
Imóvel	IdImóvel	Cada imóvel tem um número de identificação único	
Transação	idTransação	Cada transação tem um número de identificação único	

Tabela 8 - Chaves primárias e candidatas

3.6 Verificação e validação do modelo desenvolvido com as transações dos utilizadores

Visto que só existe um tipo de utilizadores, os funcionários da imobiliária terão de fazer todo o trabalho de registo e pesquisa na base de dados. Temos então que verificar quais as transações que serão mais efetuadas. São elas:

- Introduzir informações sobre imóveis, clientes e transações
- Atualizar informação sobre os imóveis
- Verificar a quantidade de imóveis à venda e por alugar
- Introduzir os dados necessário para que uma transação se efetue corretamente

3.7 Revisão e validação do modelo com o cliente

Após chegarmos a esta fase verificamos que com as entidades, já referenciadas anteriormente, conseguimos corresponder com os requisitos enunciados no capítulo 2, tal não acontecesse tínhamos que o rever novamente até se ir de encontro com essas mesmas pretensões.

Concluimos então que com o modelo conceptual elaborado anteriormente conseguimos cumprir com os requisitos pretendidos e conseguimos também responder as

perguntas elaboradas, que achamos adequadas, feitas para uma base de dados de uma imobiliária.

4 Modelação Lógica de Base de Dados

Neste capítulo, mostramos ao detalhe o processo de modelação lógico da base de dados. O objetivo desta etapa é obter um único modelo de lógico a partir do modelo conceptual anteriormente especificado, que não seja ambíguo e de fácil compreensão.

O modelo lógico traduz o modelo conceptual no modelo relacional de dados. O resultado é o conjunto de tabelas ou esquema da base de dados. Dependendo do SGBD a utilizar, o modelo lógico pode ser ligeiramente diferente, porque pode haver, por exemplo, variações nos tipos de dados, restrições no uso de nomes, tipos de índices diferentes e funcionalidades diferentes para valores por omissão ou valores automáticos. O modelo lógico permite envolver utilizadores que não sejam técnicos e que possam desta forma contribuir com especificações relativas ao problema em si, sem a necessidade de compreenderem aspetos técnicos das bases de dados [1].

Vamos então apresentar a tradução pormenorizada o modelo conceptual em lógico da nossa base de dados.

4.1 Esquema Lógico

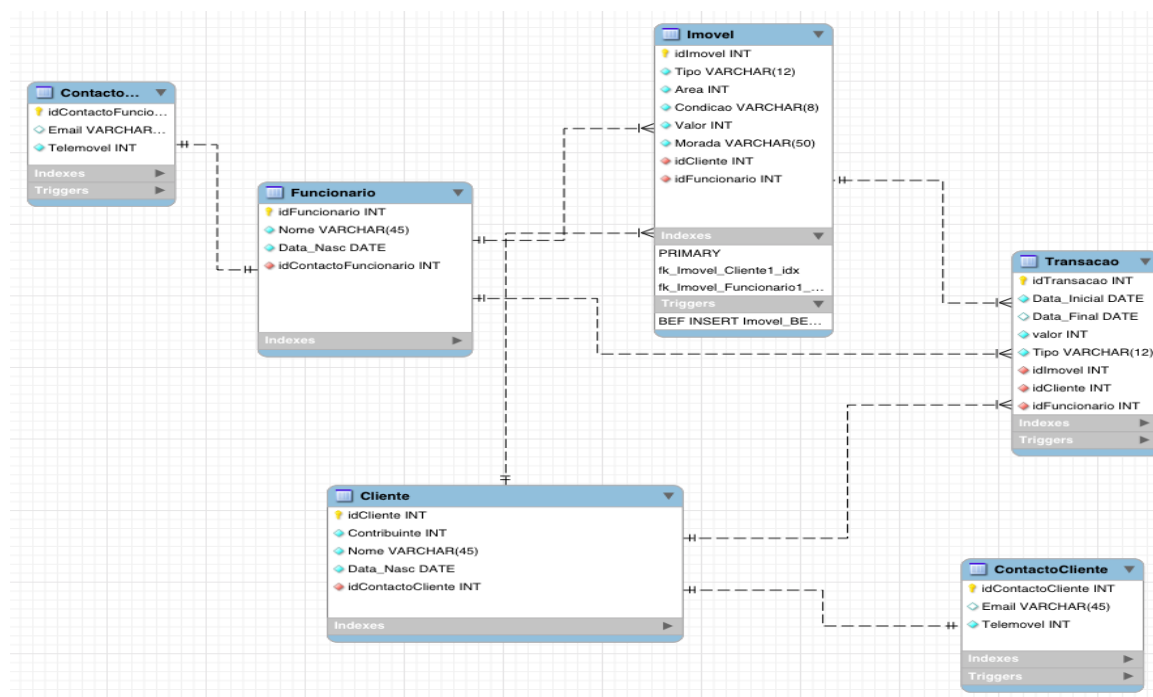


Figura 2. Modelo Lógico

4.2 Derivação de Relacionamentos para o Modelo de Dados Lógico

Nesta etapa, será descrito o processo de modelação lógica da base de dados representando as entidades, relacionamento e atributos do modelo conceptual.

4.2.1 Tipos de Entidades Fortes

Para cada entidade forte, é criada uma relação que inclui todos os atributos dessa entidade e, para cada atributo composto, apenas são incluídos os seus atributos simples.

Por exemplo, na tabela “Imóvel”, incluímos a chave primaria “idImovel” e os atributos “Morada”, “Tipo”, “Area”, “Condicao” e “Valor”.

4.2.2 Tipos de entidades fracas

No modelo construído, não existem entidades fracas uma vez que cada entidade é capaz de ser descrita por atributos próprios.

4.2.3 Relacionamentos binários 1:N

Para cada relacionamento binário 1:N, a entidade de cardinalidade 1 é chamada de entidade pai e a outra de entidade filha. Para esse tipo de relacionamento, colocamos uma cópia dos atributos das chaves primarias da entidade pai na entidade filha, atuando como chave estrangeira.

Por exemplo, no modelo construído, temos o relacionamento 1:N “Cliente ,proprietário, Imóvel”. Assim, o atributo de chave primária de “Cliente” é copiado para a tabela de “Imóvel” como chave estrangeira.

4.2.4 Relacionamentos binários 1:1

Ao contrario do relacionamento 1:N, o relacionamento 1:1 não é possível usar a cardinalidade para identificar entidade pai e entidade filha.

Usa-se então, restrições de participação para decidir se se representa o relacionamento através da combinação de entidades envolvidas numa só relação ou se se cria duas relações e copia-se a chave primaria de uma relação para a outra.

Por exemplo, no modelo construído, temos o relacionamento 1:1 “Cliente deriva em Contacto”. Este relacionamento é obrigatório no lado “Cliente” e opcional no lado “Contacto”.

Assim, “Cliente” é entidade pai e “Contacto” é entidade filha e copia-se chave primária da entidade pai para a entidade filha.

4.2.5 Remoção de Relacionamentos N:M

No processo de modelação lógica é necessário remover os relacionamentos de cardinalidade N:M, caso existam. Esta remoção é feita efetuando a divisão de cada relacionamento em dois de cardinalidade 1:N. Procedendo-se, para as tabelas resultantes, da forma descrita no capítulo 4.2.3.

No modelo construído, não existe nenhum relacionamento deste tipo.

4.2.6 Remoção de Relacionamentos Complexos

No modelo construído, não existe nenhum relacionamento deste tipo pois não existe nenhum relacionamento entre três ou mais entidades.

4.2.7 Atributos Multivalor

Atributos multivalor não têm representação direta no modelo lógico. Assim, para cada um, temos de criar uma relação para o representar e incluir a chave primária da entidade na relação recém-criada, que vai funcionar como chave estrangeira.

No modelo construído, não existe nenhum atributo deste tipo.

4.2.8 Atributos compostos

Atributos compostos são atributos formados por vários itens menores e, no modelo lógico, são representados criando uma tabela com um relacionamento 1:1 com a entidade em questão.

Por exemplo, no modelo construído, o atributo “Contacto” tem “email” e “telemovel”. Assim, cria-se uma tabela “Contacto” que tem como atributos “email” e “telemovel”.

4.3 Validação de relações através da normalização

Na etapa anterior, foi abordada a validação de atributos em cada relação usando as regras de normalização.

O processo de normalização resulta geralmente na decomposição de uma relação em várias outras, que evitam os problemas da relação original. Todos os atributos do esquema original devem estar presentes na união dos esquemas resultantes. Adicionalmente, queremos garantir que, a decomposição de uma relação em novas relações respeita a decomposição sem perdas, em que o novo esquema da base de dados contém a mesma informação do

original, ou seja, se não perder informação. Na prática, perder informação significa que, ao reconstituir-se a relação original, ganham-se tuplos que não estavam presentes inicialmente. Outra propriedade que tem de verificar é a preservação de dependências em que as novas relações devem permitir verificar as dependências funcionais que estavam presentes na relação original, caso contrario estaríamos a violar restrições fundamentais [1].

No processo de normalização somos conduzidos através de uma sequencia de passos que combinam e três etapas fundamentais, tais como Primeira Forma Normal, a Segunda Forma Normal e por fim a Terceira Forma Normal que tem com função de acabar com a redundância de dados.

Podemos dizer que o nosso projeto obedece:

Primeira Forma Normal porque todos os nossos atributos, das entidades em questão, são atômicos, isto é, não é possível decompô-los, por exemplo o imóvel contém o atributo tipo que obviamente não pode ter nesse mesmo atributo dois tipos diferentes tais como apartamento, moradia.

Uma entidade está na 2FN, se e somente se, estiver na 1FN e todos seus atributos (colunas) não chaves, dependam unicamente da chave primária. Se algum atributo depende de apenas uma parte da chave primária, isso é considerada uma violação da 2FN, logo o nosso modelo encontra-se na 2FN.

Uma entidade está na 3FN, se e somente se, estiver na 2FN e todos os atributos (colunas) não chave, forem mutuamente independentes, isto é, não há dependência funcional entre elas, e todas dependem única e exclusivamente da chave primária de forma irreduzível, concluímos então que o nosso modelo encontra-se também na 3FN.

4.4 Validação de relações contra as transações dos utilizadores

Nesta etapa, queremos verificar se as relações no modelo lógico suportam as transações exigidas.

Ao servimos da informação anteriormente recolhida sobre relações, e usando as ligações entre chaves primárias e chaves estrangeiras, o diagrama ER, tentamos executar as operações necessárias. Como resolvemos todas as transações, consequentemente validamos o modelo de dados lógico.

4.5 Verificação de restrições de integridade

As restrições de integridade são úteis para proteger a base de dados de se tornar incompleta ou inconsistente. Um modelo de dados lógico que inclua todas as restrições de integridade importantes é uma representação verdadeira dos requisitos de dados. Consideremos então, diversos tipos de restrições de integridade.

4.5.1 Restrição de Dados

Alguns atributos definidos anteriormente têm de conter obrigatoriamente um valor não nulo, como no caso do nome do funcionário, não pode existir um funcionário na imobiliária sem ter nome.

4.5.2 Restrições de domínio de atributos

Todo o atributo tem um domínio, isto é, o conjunto de valores que o atributo pode ter, por exemplo o nome do funcionário é um varchar(45).

4.5.3 Multiplicidade

Existem restrições que são colocadas nos relacionamentos entre os dados na base de dados. Um funcionário pode ter vários imóveis associados, mas um imóvel não pode ter mais do que um funcionário responsável.

4.5.4 Integridade de entidades

Uma chave primária nunca pode ser nula. Por exemplo, todos os imóveis tem de ter o valor de idImovel preenchido.

4.5.5 Integridade referencial

Uma chave estrangeira é responsável por ligar cada tuplo na relação filho ao tuplo na relação pai. A integridade referencial significa que, o valor da chave estrangeira deve referir-se a um tuplo existente na relação pai. Por exemplo, a chave "idCliente" liga as tabelas Cliente e Imovel. É necessário então que este valor não seja nulo para existir a relação.

4.6 Considerações sobre o crescimento futuro

Neste tema queremos abordar se no futuro próximo possa a ver mudanças e que o nosso modelo lógico o consiga acompanhar essas mesmas mudanças.

Devido ao facto do nosso modelo ser muito simples, conseguimos assegurar que se houver possíveis mudanças ele consegue acompanhar nos requisitos apresentados.

A imobiliária no futuro poderia trabalhar online, ou seja, tendo um site que a sustentasse e que pudesse alargar o leque de clientes.

Existem outras mudanças que poderiam ser implementadas na nossa base de dados, de modo conter mais informação (atributos) de modo a que fosse mais detalhada, mas para já consideramos contruir um projeto simples e que funcionasse, cumprisse com os requisitos especificados, para o caso em questão.

5 Modelação Física da Base de Dados

A ultima fase do nosso trabalho é a modelação física. Vamos então traduzir o modelo lógico, anteriormente especificado, no modelo físico da Base de Dados.

O modelo físico envolve a parametrização do modelo relacional em função do que se espera da sua utilização. Desta forma, tenta-se otimizar e melhorar o desempenho da base de dados. Porém ser tomadas decisões em termos de estratégias de armazenamento, estruturas de acesso rápido a dados, como os índices, ou mesmo redesenhar tabelas, para tirar partido das características de utilização previstas [1].

Após conhecermos as funcionalidades da nossa base de dados, assim como as suas vantagens e desvantagens vamos passar então a apresentar a tradução pormenorizada do modelo lógico em conceptual da nossa base de dados.

5.1 Sistema de Gestão de Base de Dados

O sistema utilizado para a realização do nosso projeto foi o *mySQL workbench*. Esta escolha baseou-se no facto de este ter sido o Sistema de Gestão de Bases de Dados utilizado nas aulas da unidade curricular de Bases de Dados.

5.2 Estruturas das relações

Para se perceber melhor o nosso modelo físico é necessário apresentarmos as relações criadas durante a modelação física da nossa Base de Dados.

Relação Funcionário:

Domínio idFuncionario	INT
Domínio Nome	VARCHAR(45)
Domínio Data_Nasc	DATE

```
`Funcionario` (  
  `idFuncionario` INT NOT NULL AUTO_INCREMENT,  
  `Nome` VARCHAR(45) NOT NULL,
```

```

`Data_Nasc` DATE NOT NULL,
`idContactoFuncionario` INT NOT NULL,
PRIMARY KEY (`idFuncionario`),
INDEX `fk_Funcionario_ContactoFuncionario1_idx`
(`idContactoFuncionario` ASC),
CONSTRAINT `fk_Funcionario_ContactoFuncionario1`
FOREIGN KEY (`idContactoFuncionario`)
REFERENCES `Imobiliaria`.`ContactoFuncionario`
(`idContactoFuncionario`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)

```

Relação Cliente:

Domínio idCliente	INT
Domínio Contribuinte	INT
Domínio Nome	VARCHAR (45)
Domínio Data_Nasc	DATE

```

`Cliente` (
  `idCliente` INT NOT NULL AUTO_INCREMENT,
  `Contribuinte` INT NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  `Data_Nasc` DATE NOT NULL,
  `idContactoCliente` INT NOT NULL,
  PRIMARY KEY (`idCliente`),
  INDEX `fk_Cliente_ContactoClientel_idx` (`idContactoCliente` ASC),
  UNIQUE INDEX `Contribuinte_UNIQUE` (`Contribuinte` ASC),
  CONSTRAINT `fk_Cliente_ContactoClientel`
    FOREIGN KEY (`idContactoCliente`)
    REFERENCES `Imobiliaria`.`ContactoCliente` (`idContactoCliente`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

```

Relação ContactoFuncionario:

Domínio Email	VARCHAR(45)
Domínio Telemovel	INT

```

`ContactoFuncionario` (
  `idContactoFuncionario` INT NOT NULL AUTO_INCREMENT,

```

```

`Email` VARCHAR(45) NULL,
`Telemovel` INT NOT NULL,
PRIMARY KEY (`idContactoFuncionario`),
UNIQUE INDEX `Telemovel_UNIQUE` (`Telemovel` ASC),
UNIQUE INDEX `Email_UNIQUE` (`Email` ASC))

```

Relação ContactoCliente:

Domínio Email	INT
Domínio Telemovel	VARCHAR (45)

```

`ContactoCliente` (
  `idContactoCliente` INT NOT NULL AUTO_INCREMENT,
  `Email` VARCHAR(45) NULL,
  `Telemovel` INT NOT NULL,
  PRIMARY KEY (`idContactoCliente`),
  UNIQUE INDEX `Telemovel_UNIQUE` (`Telemovel` ASC),
  UNIQUE INDEX `Email_UNIQUE` (`Email` ASC))

```

Relação Imóvel:

Domínio idImovel	INT
Domínio Tipo	VARCHAR (12)
Domínio Area	INT
Domínio Condicao	VARCHAR(Max 8)
Domínio Valor	INT
Domínio idCliente	INT
Domínio idFuncionario	INT
Domínio Morada	VARCHAR (50)

```

`Imovel` (
  `idImovel` INT NOT NULL AUTO_INCREMENT,
  `Tipo` VARCHAR(12) NOT NULL,
  `Area` INT NOT NULL,
  `Condicao` VARCHAR(8) NOT NULL,
  `Valor` INT NOT NULL,
  `Morada` VARCHAR(50) NOT NULL,
  `idCliente` INT NOT NULL,

```

```

`idFuncionario` INT NOT NULL,
PRIMARY KEY (`idImovel`),
INDEX `fk_Imovel_Cliente1_idx` (`idCliente` ASC),
INDEX `fk_Imovel_Funcionario1_idx` (`idFuncionario` ASC),
CONSTRAINT `fk_Imovel_Cliente1`
    FOREIGN KEY (`idCliente`)
    REFERENCES `Imobiliaria`.`Cliente` (`idCliente`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_Imovel_Funcionario1`
    FOREIGN KEY (`idFuncionario`)
    REFERENCES `Imobiliaria`.`Funcionario` (`idFuncionario`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

```

Relação Transação:

Domínio idTransacao	INT
Domínio Data Inicial	DATE
Domínio Data Final	DATE
Domínio Tipo	VARCHAR(12)
Domínio idImovel	INT
Domínio idCliente	INT
Domínio idFuncionario	INT
Domínio valor	INT

```

`Transacao` (
    `idTransacao` INT NOT NULL AUTO_INCREMENT,
    `Data_Inicial` DATE NOT NULL,
    `Data_Final` DATE NULL,
    `valor` INT NOT NULL,
    `Tipo` VARCHAR(12) NOT NULL,
    `idImovel` INT NOT NULL,
    `idCliente` INT NOT NULL,
    `idFuncionario` INT NOT NULL,
    PRIMARY KEY (`idTransacao`),
    INDEX `fk_Transacao_Imovell_idx` (`idImovel` ASC),
    INDEX `fk_Transacao_Cliente1_idx` (`idCliente` ASC),
    INDEX `fk_Transacao_Funcionario1_idx` (`idFuncionario` ASC),
    CONSTRAINT `fk_Transacao_Imovell`
        FOREIGN KEY (`idImovel`)

```

```

REFERENCES `Imobiliaria`.`Imovel` (`idImovel`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Transacao_Cliente1`
FOREIGN KEY (`idCliente`)
REFERENCES `Imobiliaria`.`Cliente` (`idCliente`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Transacao_Funcionario1`
FOREIGN KEY (`idFuncionario`)
REFERENCES `Imobiliaria`.`Funcionario` (`idFuncionario`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)

```

5.3 Restrições gerais

Nesta fase são definidas as restrições gerais aplicadas ao mundo real, dos dados armazenados, por exemplo não será possível a venda de um imóvel a um preço inferior ou igual a 0.

Relação Imóvel

- A condição do Imóvel só pode ter quatro valores 'Alugado', 'Aluguer', 'Venda', 'Vendido'.
- O tipo do Imóvel só pode ter os valores 'Terreno', 'Moradia', 'Apartamento', 'Loja'.
- O preço do Imóvel não pode ser inferior ou igual a 0.
- A área do Imóvel não pode ser inferior ou igual a 0.

```

CREATE          DEFINER          =          CURRENT_USER          TRIGGER
`Imobiliaria`.`Imovel_BEFORE_INSERT` BEFORE INSERT ON `Imovel` FOR
EACH ROW
BEGIN
    IF (NEW.Condicao NOT LIKE 'Alugado') OR
       (NEW.Condicao NOT LIKE 'Aluguer') OR
       (NEW.Condicao NOT LIKE 'Vendido') OR
       (NEW.Condicao NOT LIKE 'Venda') THEN
        SIGNAL SQLSTATE '45000'

```

```

        SET MESSAGE_TEXT = "Condição não conhecida"
    END IF;

    IF (NEW.Tipo NOT LIKE 'Moradia') OR
        (NEW.Tipo NOT LIKE 'Apartamento') OR
        (NEW.Tipo NOT LIKE 'Loja') OR
        (NEW.Tipo NOT LIKE 'Terreno') THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Tipo não conhecido"
    END IF;

    IF NEW.Valor<=0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Valor não conhecido"
    END IF;

    IF NEW.Area<=0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Area não conhecido"
    END IF;
END

```

Relação Transação

- O preço da transação não pode ser inferior ou igual a 0.
- A data de início da transação não pode ser inferior a data final.
- O tipo de transação só pode ter os valores 'Aluguer' e 'Venda'

```

CREATE          DEFINER          =          CURRENT_USER          TRIGGER
`Imobiliaria`.`Transacao_BEFORE_INSERT` BEFORE INSERT ON `Transacao`
FOR EACH ROW
BEGIN
    IF NEW.valor<=0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Valor não conhecido"
    END IF;

    IF (NEW.Tipo NOT LIKE 'Moradia') OR
        (NEW.Tipo NOT LIKE 'Apartamento') OR
        (NEW.Tipo NOT LIKE 'Loja') OR
        (NEW.Tipo NOT LIKE 'Terreno') THEN
        SIGNAL SQLSTATE '45000'
    END IF;
END

```

```

        SET MESSAGE_TEXT = "Tipo não conhecido"
    END IF;

    IF DATEDIFF(NEW.Data_Final,NEW.Data_Inicial)<365 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Tipo não conhecido"
    END IF;
END

```

Relação ContactoCliente

- O email do Cliente tem de ser válido caso exista.

```

CREATE DEFINER = CURRENT_USER TRIGGER
`Imobiliaria`.`ContactoCliente_BEFORE_INSERT` BEFORE INSERT ON
`ContactoCliente` FOR EACH ROW
BEGIN
    IF (NEW.Email not like '%@%.%' or NEW.Email is not null) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Formato do email incorreto"
    END IF;
END

```

Relação ContactoFuncionario

- O email de um Funcionário tem de ser válido caso exista.

```

CREATE DEFINER = CURRENT_USER TRIGGER
`Imobiliaria`.`ContactoFuncionario_BEFORE_INSERT` BEFORE INSERT ON
`ContactoFuncionario` FOR EACH ROW
BEGIN
    IF (NEW.Email not like '%@%.%' or (NEW.Email is not null)) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Formato do email incorreto"
    END IF;
END

```

5.4 Organização de ficheiros e índices

Nesta secção é analisada a forma como são guardados os ficheiros que constituem o sistema de base de dados. É também analisada a necessidade de criação de índices no sentido de aumentar a performance do sistema.

5.4.1 Análise de Transações

Depois de uma análise das transações entre relações, da identificação das mais importantes e da elaboração das principais queries verifica-se que a maioria das operações são de consulta e inserção de informação, existindo também algumas de alteração.

Algumas transações possíveis, no modelo construído:

1. Inserir um novo imóvel:

Para inserir um novo imóvel devemos verificar se ele já existe na base de dados. Se não existir, é criado um imóvel. Se já existir e não estiver marcado para venda, simplesmente altera-se esse “marcador” e o novo preço pretendido.

2. Inserir um novo cliente:

Para inserir um novo cliente devemos verificar se ele já existe na base de dados. Se não existir, é criado um novo cliente. Se já existir, não se faz mais nada.

3. Selecionar todos os imoveis com um valor igual ou superior a ‘x’, para venda, em Braga:

Para realizar esta transação temos que listar todos os imoveis da tabela de imoveis com os campos “Condicao” = “Venda”, “Morada”=“Braga” e “valor” maior ou igual a ‘x’.

5.5 Resposta às perguntas de validação feitas pelo utilizador

- 1) É possível adicionar imóveis à base de dados?
- 2) É possível criar (adicionar) ficha cliente?
- 3) É possível adicionar um novo funcionário?
- 4) Qual funcionário foi/está responsável por determinada transação?
- 5) Quantos imóveis estão à venda?
- 6) Quais imóveis estão disponíveis para alugar?
- 7) Quais imóveis estão vendidos?
- 8) Quais imóveis estão alugados?
- 9) Qual o valor de determinado imóvel?
- 10) Quais imóveis pertencentes a uma certa cidade?

--1

```
insert into Imovel (Morada,
Tipo,Area,Condicao,Valor,idCliente,idFuncionario)
values ('Caminho da Pereira, n284,
Porto','Apartamento',364,'Venda',364000,67,10);
```

--2

```
insert into ContactoCliente (Email,Telemovel)
values ('aline@live.co.pt',933258803);
insert into Cliente (Contribuinte,Nome,idContactoCliente)
values (480116752,'Aline Paiva',1);
```

--3

```
insert into ContactoFuncionario (Email,Telemovel)
values ('ronny@hotmail.com',915350056);
insert into Funcionario (Nome,idContactoFuncionario)
values ('Ronny Tavares',1);
```

-- 4

```
SELECT F.idFuncionario as CodigoFuncionario, F.Nome
FROM Transacao as T
INNER JOIN Funcionario as F
ON T.idFuncionario = F.idFuncionario
```

```
WHERE T.idTransacao = 1;
```

-- 5

```
Select Count(idImovel)
      From Imovel
      WHERE Condicao = 'Venda';
```

-- 6

```
Select Count(idImovel)
      From Imovel
      WHERE Condicao = 'Aluguer';
```

-- 7

```
Select Count(idImovel)
      From Imovel
      WHERE Condicao = 'Vendido';
```

-- 8

```
Select Count(idImovel)
      From Imovel
      WHERE Condicao = 'Alugado';
```

-- 9

```
Select Valor
      From Imovel
      WHERE idImovel = 1;
```

-- 10

```
Select idImovel asCodigo , Morada , Valor
      From Imovel
      Where Morada like ('%Braga%') and valor >= 50000;
```

5.6 Escolha e criação de índices

Neste capítulo vamos abordar a temática dos índices que é uma parte fundamental de uma base de dados. Em muitas situações de pesquisa na BD dizem respeito a valores específicos com atributos em questão, e se queremos fazer essa pesquisa não queremos naturalmente que ela “varra” toda a base de dados para procurar esse atributo.

Por isso é que vamos criar índices, apesar de afetar o espaço em disco, resolvemos cria-los porque a nossa base de dados do projeto da imobiliária é relativamente pequena cerca de 20 funcionários e 100 clientes. Nesse sentido decidimos então que o contribuinte seria o índice para que assim se pudesse pesquisar um cliente, pois ele não saberá o idCliente que corresponde a identificação do cliente na BD, assim fica então facilitada a pesquisa através desse atributo de um determinado cliente.

5.7 Análise de espaço em disco

Para este projeto utilizamos o motor de armazenamento InnoDB do MySQL. Precisamos então de perceber qual é o espaço que cada tipo utilizado na construção da base de dados ocupa.

INT: 4 bytes
DATE: 3 bytes
VARCHAR(N): N+1 bytes

Após efetuar os cálculos, e tendo em conta a informação em cima, chegamos aos seguintes valores:

ContactoFuncionario	$45 \cdot 20 + 4 \cdot 20$	980 bytes
ContactoCliente	$45 \cdot 100 + 4 \cdot 100$	4900 bytes
Cliente	$4 \cdot 100 + 46 \cdot 100 + 3 \cdot 100 + 4 \cdot 100$	5700 bytes
Imovel	$46 \cdot 100 + 13 \cdot 100 + 4 \cdot 100 + 9 \cdot 100 + 4 \cdot 100 + 4 \cdot 100 + 4 \cdot 100$	8400 bytes
Funcionario	$46 \cdot 20 + 3 \cdot 20 + 4 \cdot 20$	1060 bytes
Transacao	$3 \cdot 194 + 3 \cdot 194 + 4 \cdot 194 + 4 \cdot 194 + 4 \cdot 194 + 4 \cdot 194$	4268 bytes
TOTAL		25308 bytes

5.8 Mecanismos de segurança

Nesta etapa iremos explicar os mecanismos de segurança que afetam os utilizadores, no nosso caso só temos um tipo de utilizador, que será o funcionário da imobiliária.

Este terá controlo sobre a base de dados, tem a capacidade de adicionar imoveis assim como clientes, realizar transações, mas não poderá remover outros funcionários, imoveis e transações, mas terá capacidade de alterar os dados dos imoveis.

Cria Utilizador

```
CREATE USER 'empregado'@'localhost' IDENTIFIED BY 'chave';
```

Adicionar Permissões

```
GRANT ALL ON Imovel to 'empregado'@'localhost';
```

```
GRANT ALL ON Cliente to 'empregado'@'localhost';
```

```
GRANT ALL ON Funcionario to 'empregado'@'localhost';
```

```
GRANT ALL ON Transacao to 'empregado'@'localhost';
```

Remover Permissões

```
REVOKE DELETE ON Imovel to 'empregado'@'localhost';
```

```
REVOKE DELETE ON Cliente to 'empregado'@'localhost';
```

```
REVOKE DELETE ON Funcionario to 'empregado'@'localhost';
```

```
REVOKE DELETE ON Transacao to 'empregado'@'localhost';
```

6 Ferramentas utilizadas

No desenvolvimento deste trabalho utilizamos às seguintes ferramentas:

- Microsoft Word
- TerraER: Desenho do diagrama ER e do modelo conceptual.
- MySQL Workbench: Desenvolvimento do modelo lógico e físico.

7 Conclusão

“Toda a teoria deve ser feita para poder ser posta em prática, e toda a prática deve obedecer a uma teoria. “ - Fernando Pessoa

No decorrer deste trabalho, foi-nos dada a oportunidade de modelar uma Base de Dados, possibilitando-nos uma melhor compreensão de todo o processo. Compreensão essa que só é possível “pondo as mãos na massa”, não pondo de parte a base teórica que é preciso.

Tendo isto em mente, o que retiramos deste trabalho é que para conseguirmos chegar a uma Base de Dados bem executada é preciso pensar bem o problema, ou seja, temos de pensar qual é o problema que queremos suprimir para passarmos para a análise de requisitos e contacto direto com o cliente, ou neste caso, entre o grupo, perceber o que é mesmo necessário, o que está em falta e o que está a mais. Percebemos também a importância da realização dos modelos (conceptual, lógico e físico), que pareciam muito teóricos, mas são uma parte muito importante do processo da criação da Base de Dados e nos ajudam a remover redundâncias, no caso do modelo conceptual, que após se ponderar se está de acordo com os requisitos é convertido no modelo lógico e finalmente no físico.

Entendemos que há muito a melhorar nesta base de dados se fosse para ser aplicada no mundo real, mas na nossa perspetiva há sempre, seja em termos de performance, ou até da sua estrutura. Porém este pequeno trabalho é só uma amostra do que pode ser realizado com o que é lecionado na Unidade Curricular de Base de Dados.

Em suma, apesar de ser um trabalho bastante simples, conseguimos perceber que a teoria anda de mão dada com a prática e no mundo das Bases de Dados não é possível chegar a bom porto sem a conjugação das duas. Percebemos também que Fernando Pessoa é um génio intemporal, mas isso é conversa para outra altura.

8 Referências

[1] Gouveia, F., "Fundamentos de Bases de Dados", Coleção Tecnologias da Informação, FCA, Editora de Informática, 2014.

9 Lista de Siglas e Acrónimos

BD Base de Dados
ER Entidade e Relacionamento
SQL Structured Query Language
1FN Primeira forma Normal
2FN Segunda Forma Normal
3FN Terceira Forma Normal