



Universidade do Minho
Licenciatura em Engenharia Informática

Unidade Curricular de Bases de Dados

Ano Lectivo de 2022/2023

LTR Imobiliária: Sistema de Gestão de Bases de Dados

BD

João Pedro Baptista a100705
João Pedro Rodrigues a1000896
Mateus Martins a100645
António Pedro a100821

Junho - 2023

Data de Recepção	
Responsável	
Avaliação	
Observações	

LTR Imobiliária: Sistema de Gestão de Bases de Dados

João Pedro Baptista a100705
 João Pedro Rodrigues a1000896
 Mateus Martins a100645
 António Pedro a100821

Junho - 2023

Índice

DEFINIÇÃO DE SISTEMA	3
1. Contextualização.....	3
2. Fundamentação	3
3. Objetivos.....	3
4. Viabilidade	3
5. Recursos Humanos	4
6. Equipa de Trabalho	4
7. Plano de Execução	4
8. Revisão e Aprovação.....	5
 DEFINIÇÃO DE REQUISITOS	 5
1. Método de Levantamento	5
2. Levantamento	5
3. Requisitos de Descrição	5
4. Requisitos de Manipulação	6
5. Requisitos de Controlo	7
 Modelo Conceptual	 7
1. Identificação e caracterização das entidades	8
2. Identificação e caracterização dos relacionamentos.....	8
3. Identificação e caracterização dos atributos	8
 Modelo Lógico.....	 9
1. Apresentação e explicação do modelo lógico produzido	9
2. Validação e Normalização dos Dados	9
 Sistema de Recolha de Dados	 10
1. Migração de dados dos CSVs para o MYSQL (Python).....	10
 Implementação Física	 11
1. Implementação da base de dados no MYSQL	11
2. Queries desenvolvidas para dar informações aos utilizadores	11
 Conclusão.....	 13
Bibliografia	14

DEFINIÇÃO DE SISTEMA

1. Contextualização

A LRT é uma multinacional portuguesa, fundada na cidade de Guimarães em Julho de 1953, que iniciou a sua atividade na área da construção civil, área em ascensão no pós-guerra.

Após quase 4 décadas, tornou-se líder destacada no mercado mundial em 1991, consequência da política inovadora e vanguardista da empresa, que lhe permitiu a criação de técnicas de construção mais eficientes e ecológicas.

Fruto de longos anos de contacto com outras áreas e com o intuito de aumentar os lucros da empresa, decidiu diversificar as suas áreas de atuação, e na linha do ecologismo anteriormente seguido, decidiu investir nas energias renováveis, seguidamente, e com o avanço das novas tecnologias a área da robótica foi visada, permitindo também a automação da parte inicial de construção civil.

Após a última reunião da Administração, a mesma decidiu que dada a crise de habitação que o país enfrenta o mercado de imóveis seria um bom próximo investimento.

2. Fundamentação

Com o objetivo em mente e tendo em conta a dimensão de tal projeto, a experiência da administração fê-los achar necessária a criação de uma base de dados para melhor gestão dos imóveis que já se encontram em sua posse e para monitorização daqueles que pretendem adquirir.

3. Objetivos

Baseada nos sucessos obtidos nas áreas em que a empresa já se encontra estabelecida a Administração apontou os objetivos que pretende atingir com a implementação do desejado sistema de bases de dados:

- Monitorizar o valor de mercado e outras características dos seus imóveis e daqueles que desejam comprar;
- Gerir possíveis inquilinos para esses imóveis;
- Calcular estatísticas que permitam saber a situação do mercado;
- Ter assim que entrem em atividade um sistema que permita uma gestão automática das finanças desta divisão da empresa.

4. Viabilidade

Tendo em conta a já existência de SBDs noutras divisões da empresa e a dimensão do investimento nesta nova área, a empresa pensa que a implementação de um SBD para gestão da mesma é essencial pois:

- Facilitará a integração desta nova área nos sistemas informáticos da empresa, facilitando simultaneamente a gestão da mesma;
- Permitirá um retorno de cerca de 3000% sobre o investimento no SBD, após um mês da sua implementação.

5. Recursos Humanos

Funcionários da empresa, agentes imobiliários, arrendatários, proprietários, e funcionários da empresa de desenvolvimento.

Materiais: Hardware e Software

6. Equipa de Trabalho

- Pessoal Interno:

João Felgueiras, diretor do Departamento de Informático da LTR, responsável pela integração deste projeto com o sistema já existente na empresa.

Hélio Fernandes, responsável na empresa pela aquisição dos imóveis.

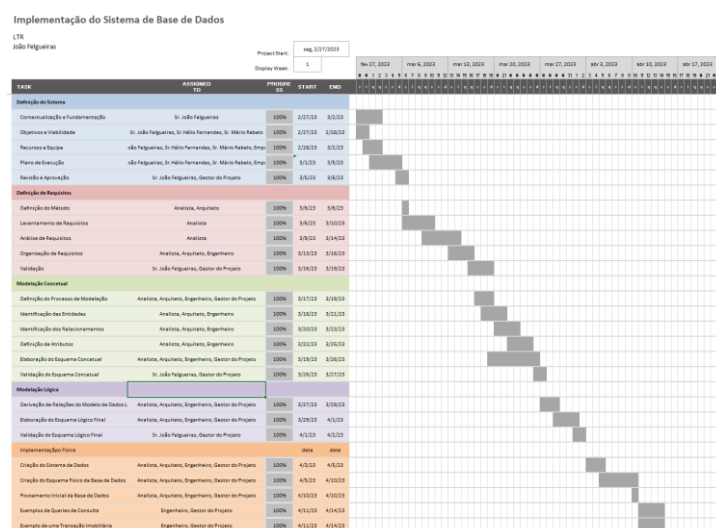
Mário Rebelo, ex-agente imobiliário que em coordenação com Hélio Fernandes gere os imóveis em posse e aqueles que a empresa deve visar.

- Pessoal Externo:

Equipa de 4 elementos da empresa a contratar, composta por um gestor de projeto, um analista e arquitetos e engenheiros de Bases de Dados, responsáveis por levantar os requisitos, modelar e implementar o sistema.

7. Plano de Execução

Após uma reunião entre o Sr. João Felgueiras e o gestor do projeto da empresa contratada, foi definido um plano de trabalhos, bem estruturado, que incorpora todas as fases do ciclo de vida de um SBD, distribuindo já a execução de diferentes etapas pelos integrantes da equipa de trabalho, bem como a calendarização dos mesmos, tendo toda essa informação sido representada no diagrama de GANTT.



8. Revisão e Aprovação

Nessa mesma reunião, depois da apresentação do diagrama de GANTT, toda a proposta foi de novo analisada, tendo sido esclarecidas algumas questões quanto a funcionalidades da aplicação, à integração da mesma junto dos sistemas já existentes e à capacidade do Sistema receber novas funcionalidades, caso necessário. Sem necessidade de alterações ao que havia sido definido anteriormente, o plano para implementação do SBD foi aprovado pelo Sr. João Felgueiras, tendo a empresa contactada sido contratada.

DEFINIÇÃO DE REQUISITOS

1. Método de Levantamento

Devido a tratar-se de uma nova divisão que está a ser criada na empresa, os requisitos de funcionalidades para o SBD, provieram exclusivamente de reuniões com a equipa de trabalho interno referida anteriormente, onde foram identificadas todas as necessidades que a empresa tem, e as características que o SBD deve possuir para responder a essas necessidades.

2. Levantamento

Nessas reuniões foi identificada apenas uma área de implementação do sistema, a manutenção de um registo das informações relativas às propriedades que estejam a ser observadas pela empresa para aquisição ou as que já estão em posse da mesma.

Para cada um dos requisitos identificados, foram também registados no documento de requisitos, quem providenciou o requisito, quem procedeu ao seu registo, a data e hora do levantamento, a área de utilização, bem como detalhes sobre o requisito em si.

3. Requisitos de Descrição

Nº	Descrição	Área	Data e Hora	Fonte
RD1	A cada propriedade deve ser atribuído um ID de 8 algarismos que deve ser convertido para hexadecimal posteriormente.	Registo das propriedades	06/03/2023 09:46	Sr. João Felgueiras
RD2	É necessário armazenar os dados de cada propriedade, relativos à sua área, valor atual, localização e utilização.	Registo das propriedades	06/03/2023 09:48	Sr. Hélio Fernandes, Sr. Mário Rebelo

RD3	Deve também ser armazenada informação relativas a contratos em vigor no presente ou que tenham sido assinados no passado, tanto de compra e venda, como de arrendamento.	Registo das propriedades	06/03/2023 09:53	Sr. Hélio Fernandes, Sr. Mário Rebelo
RD4	Para os contratos de compra e venda deve ser armazenada informação relativa ao anterior e ao novo proprietário, a propriedade, data de assinatura e valor do negócio. Deve também ser atribuído um ID.	Registo das propriedades	08/03/2023 10:09	Sr. Hélio Fernandes, Sr. Mário Rebelo
RD5	Para os contratos de arrendamento deve ser armazenada informação relativa à propriedade, arrendatário, data de assinatura, duração do contrato e valor mensal. Deve também ser atribuído um ID.	Registo das propriedades	08/03/2023 10:25	Sr. Hélio Fernandes, Sr. Mário Rebelo
RD6	Para os proprietários e arrendatários deve ser armazenada a informação relativa ao seu nome, data de nascimento, género, profissão e área de residência. Deve também ser atribuído um ID.	Registo das propriedades	08/03/2023 10:37	Sr. Hélio Fernandes, Sr. Mário Rebelo

4. Requisitos de Manipulação

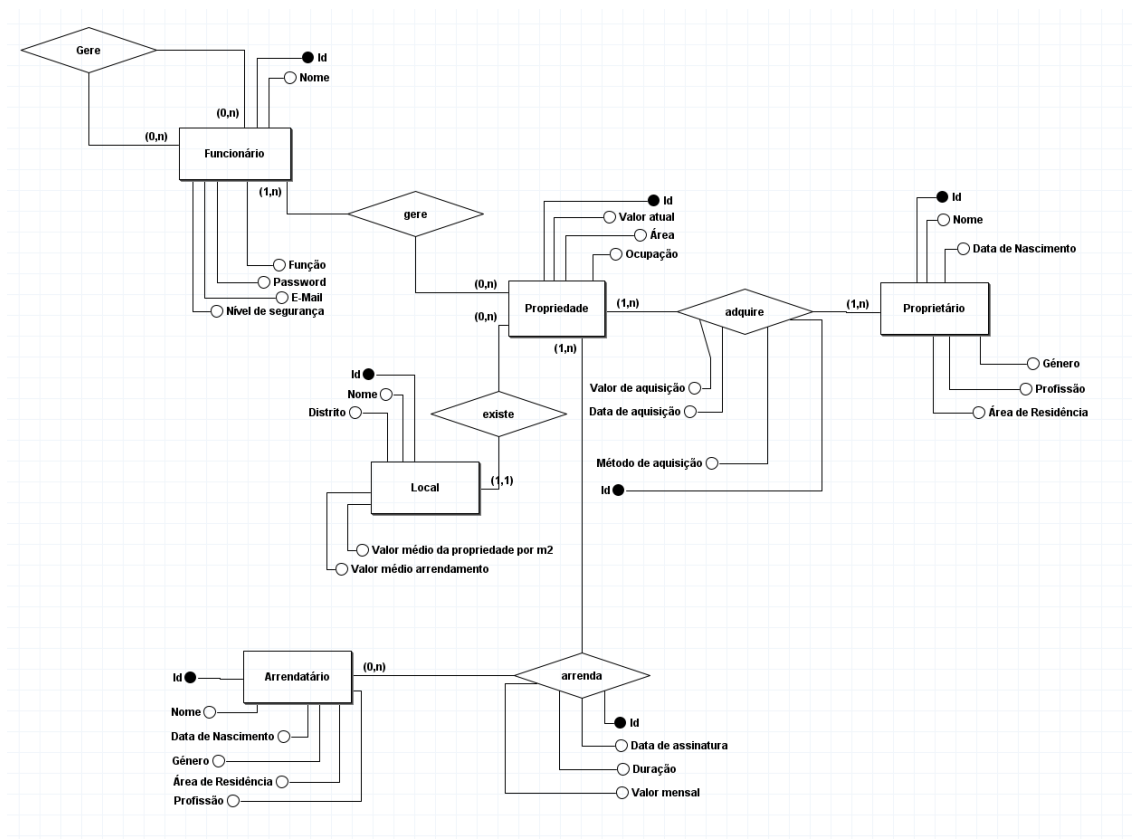
Nº	Descrição	Área	Data e Hora	Fonte
RM1	Deve existir a funcionalidade de adicionar e remover propriedades ao SBD.	Registo das propriedades	06/03/2023 09:40	Sr. João Felgueiras
RM2	Deve existir a funcionalidade de adicionar e remover proprietários e arrendatários ao SBD.	Registo das propriedades	08/03/2023 10:47	Sr. João Felgueiras
RM3	Deve existir a funcionalidade de adicionar e remover contratos de arrendamento e compra e venda ao SBD.	Registo das propriedades	08/03/2023 10:53	Sr. João Felgueiras
RM4	Obter uma lista com as n propriedades mais valiosas a serem observadas.	Registo das propriedades	10/03/2023 16:13	Sr. João Felgueiras, Sr. Hélio Fernandes, Sr. Mário Rebelo
RM5	Obter uma lista com as n propriedades que têm o menor ou maior valor de mercado por m2, no geral.	Registo das propriedades	10/03/2023 16:23	Sr. João Felgueiras, Sr. Hélio Fernandes, Sr. Mário Rebelo
RM6	Obter o preço médio por m2 e de arrendamento por m2 e o valor médio de mercado.	Registo das propriedades	10/03/2023 16:38	Sr. João Felgueiras, Sr. Hélio Fernandes, Sr. Mário Rebelo
RM7	Calcular o valor recebido da empresa proveniente das suas propriedades arrendadas.	Registo das propriedades	10/03/2023 16:43	Sr. João Felgueiras, Sr. Hélio Fernandes, Sr. Mário Rebelo
RM8	Obter o Local onde se situa a propriedade mais valiosa no momento.	Registo das propriedades	10/03/2023 16:50	Sr. João Felgueiras, Sr. Hélio Fernandes, Sr. Mário Rebelo

RM9	Calcular a % da distribuição das riquezas no mercado imobiliário, por género do proprietário.	Registo das propriedades	10/03/2023	Sr. João Felgueiras, Sr. Hélio Fernandes, Sr. Mário Rebelo
-----	---	--------------------------	------------	--

5. Requisitos de Controlo

Nº	Descrição	Área	Data e Hora	Fonte
RC1	Devem existir 3 níveis de acesso ao SBD: 1, 2 e 3.	Registo das propriedades	06/03/2023 09:11	Sr. João Felgueiras
RC2	Os utilizadores de nível 1 apenas devem poder consultar as informações relativas a propriedades no SBD	Registo das propriedades	06/03/2023 09:15	Sr. João Felgueiras
RC3	Os utilizadores de nível 2 têm acesso a toda a informação do SBD, tanto de propriedades como utilizadores, e podem manipular totalmente os dados relativos a propriedades.	Registo das propriedades	06/03/2023 09:22	Sr. João Felgueiras
RC4	Os utilizadores de nível 3 têm total acesso ao SBD podendo aceder e alterar tanto os dados das propriedades como dos utilizadores.	Registo das propriedades	06/03/2023 09:35	Sr. João Felgueiras

Modelo Conceptual



1. Identificação e caracterização das entidades

- Funcionário: Esta entidade representa um funcionário da LTR, podendo ser um agente Imobiliário, Supervisor, entre outros;
- Proprietário: Um proprietário representa uma pessoa que tem ou pode vir a possuir, vender e arrendar propriedades;
- Arrendatário: Um arrendatário é também uma pessoa que representa um inquilino.
- Local: Um local é, como o nome indica, uma Cidade, Freguesia, etc.
- Propriedade: Esta é a entidade que caracteriza um imóvel.

2. Identificação e caracterização dos relacionamentos

- gere: Um funcionário pode gerir nenhum, um ou vários, funcionando assim como supervisor. Outro relacionamento que utiliza esse nome é entre um funcionário e uma propriedade. Um funcionário pode gerir 0, uma ou várias propriedades, porém uma propriedade tem de ser gerida por 1 ou vários funcionários.
- existe: Uma propriedade existe num único local, porém num local existem 0, uma ou várias propriedades.
- arrenda: Este relacionamento representa um Contrato de Arrendamento de uma Propriedade. Uma propriedade pode ser arrendada a nenhum, um ou vários arrendatários. Pode ser arrendada a vários pelo único motivo que a Base de Dados guarda os dados de arrendamentos anteriores. Um arrendatário pode arrendar uma ou várias propriedades.
- adquire: Este relacionamento representa um Contrato de Compra de uma Propriedade. Uma propriedade pode ser adquirida por 1 ou vários proprietários (devido à salvaguarda dos antigos proprietários), enquanto que um proprietário pode adquirir 1 ou várias propriedades.

3. Identificação e caracterização dos atributos

Cada entidade tem o seu próprio identificador (ID) que permite utilizar as mesmas como chaves estrangeiras noutras tabelas.

Um proprietário e arrendatário possuem também como atributos: Nome, Data de Nascimento, Género, Profissão e Área de Residência.

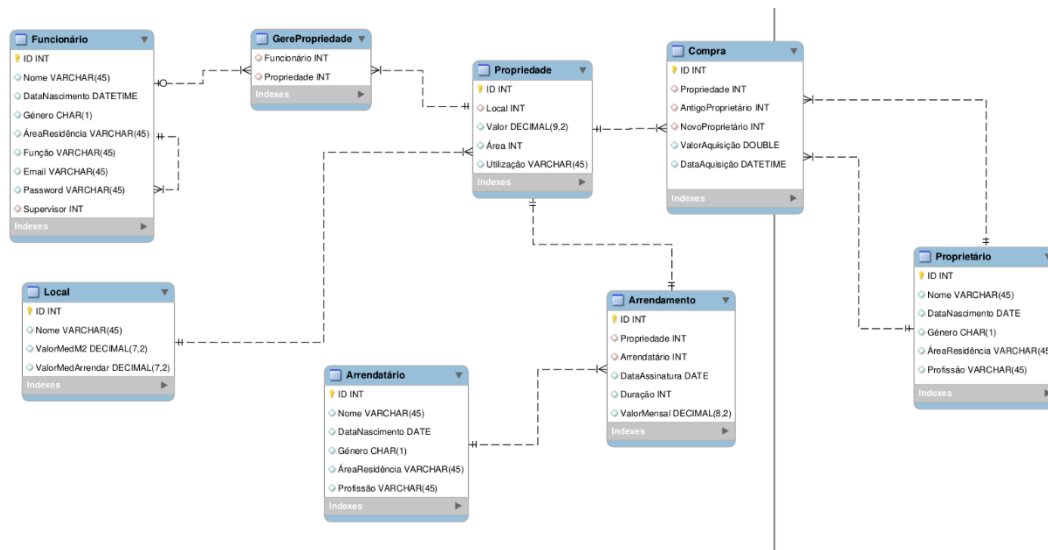
Já um funcionário possui também: Nome, Função, Password e Email e Nível de Segurança.

Um local tem guardado o seu nome, distrito, o valor médio por m² e o valor médio de arrendamento.

A propriedade tem também o valor atual da mesma, a área e a ocupação.

Apenas os relacionamentos “arrenda” e “adquire” é que possuem atributos. Os mesmos possuem ID e informações relativas a esse único arrendamento e compra, tais como: Data de assinatura de contrato, duração e valor mensal para os arrendamentos, e valor de aquisição, data e método para as compras.

Modelo Lógico



1. Apresentação e explicação do modelo lógico produzido

Para a criação do modelo lógico começamos por criar as tabelas de cada uma das entidades, com os respectivos atributos, sendo estas a "Funcionário", "Propriedade", "Proprietário", "Arrendatário" e "Local". É de notar que cada uma destas tabelas tem como chave primária um ID para que possam ser representadas sem erros de ambiguidade. Para além desse ID, cada tabela pode conter como chaves estrangeiras atributos provenientes de relacionamentos.

Posteriormente são criadas tabelas necessárias devido aos relacionamentos. Estas são a "GerePropriedade", "Compra" e "Arrendamento".

A "GerePropriedade" apenas contém duas chaves estrangeiras, sendo estas o ID de um funcionário e o ID de uma propriedade gerida pelo mesmo.

A tabela "Compra" representa um contrato de compra de uma propriedade, tendo como chave primária um ID e como chaves estrangeiras os IDs da propriedade, do antigo e novo proprietários. Além destes também tem como atributo o valor e data de aquisição da propriedade.

Já a tabela "Arrendamento" representa um contrato de arrendamento, possuindo também como chave primária um ID. As suas chaves estrangeiras são os identificadores da propriedade e arrendatário em questão. Como atributos tem também a data de assinatura, a duração de contrato e o respetivo valor mensal.

2. Validação e Normalização dos Dados

Poderemos verificar para a validação elementos como: a correspondência com os requisitos propostos, a representação correta dos relacionamentos, a integridade dos dados, entre outros.

Passemos agora à verificação de normalização de dados:

Todo corpo do modelo respeitam as formas normais. Podemos verificar tal afirmação devido à existência de chaves primárias como identificadores (1ª forma normal), à dependência dos atributos não chave relativamente às respetivas chaves primárias (2ª forma normal), e à independência dos atributos não chave entre si (3ª forma normal).

Sistema de Recolha de Dados

1. Migração de dados dos CSVs para o MYSQL (Python)

Para migrar os dados dos ficheiros CSV para o MYSQL, optamos por fazer um pequeno script em Python que lê os mesmos (utilizando a biblioteca “csv”) e, utilizando a biblioteca mysql.connector, conecta-se ao servidor localhost para ter acesso às bases de dados. Um excerto do código está apresentado abaixo:

```
import csv
import mysql.connector

def create_connection():
    # Replace with your actual database connection details
    connection = mysql.connector.connect(
        host='localhost',
        user='root',
        password='<password>',
        database='LTR'
    )
    return connection

# Function to insert data into the 'Funcionário' table
def insert_funcionarios(connection):
    with open('Funcionários.csv', 'r') as file:
        reader = csv.DictReader(file, delimiter=';')

        cursor = connection.cursor()

        for row in reader:
            id = int(row['ID'])
            nome = row['Nome']
            data_nascimento = row['DataNascimento']
            genero = row['Género']
            area_residencia = row['ÁreaResidência']
            funcao = row['Função']
            email = row['Email']
            password = row['Password']
            supervisor = int(row['Supervisor']) if row['Supervisor'] else 'NULL'

            query = f"INSERT INTO `Funcionário` (`ID`, `Nome`, `DataNascimento`, `Género`, `ÁreaResidência`, `Função`, `Email`, `Password`, `Supervisor`) VALUES ({id}, '{nome}', '{data_nascimento}', '{genero}', '{area_residencia}', '{funcao}', '{email}', '{password}', {supervisor})"

            cursor.execute(query)

        connection.commit()
        print("Data inserted into 'Funcionário' table successfully.")
```

Implementação Física

1. Implementação da base de dados no MYSQL

Para criar as tabelas e a base de dados, fizemos um script SQL mostrado abaixo:

```
CREATE DATABASE IF NOT EXISTS LTR;

CREATE TABLE IF NOT EXISTS `LTR`.`Funcionário` (
  `ID` INT NOT NULL,
  `Nome` VARCHAR(45) NULL,
  `DataNascimento` DATE NULL,
  `Gênero` CHAR(1) NULL,
  `ÁreaResidência` VARCHAR(45) NULL,
  `Função` VARCHAR(45) NULL,
  `Email` VARCHAR(45) NULL,
  `Password` VARCHAR(45) NULL,
  `Supervisor` INT NULL,
  PRIMARY KEY (`ID`),
  INDEX `Supervisor_idx` (`Supervisor` ASC) VISIBLE,
  CONSTRAINT `Supervisor`
    FOREIGN KEY (`Supervisor`)
      REFERENCES `LTR`.`Funcionário` (`ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

CREATE TABLE IF NOT EXISTS `LTR`.`Local` (
  `ID` INT NOT NULL,
  `Nome` VARCHAR(45) NULL,
  `ValorMedioQ` DECIMAL(7,2) NULL,
  `ValorMedioRenda` DECIMAL(7,2) NULL,
  PRIMARY KEY (`ID`));

CREATE TABLE IF NOT EXISTS `LTR`.`Propriedade` (
  `ID` INT NOT NULL,
  `Local` INT NULL,
  `Valor` DECIMAL(9,2) NULL,
  `Área` INT NULL,
  `Utilização` VARCHAR(45) NULL,
  PRIMARY KEY (`ID`),
  INDEX `Local_idx` (`Local` ASC) VISIBLE,
  CONSTRAINT `Local`
    FOREIGN KEY (`Local`)
      REFERENCES `LTR`.`Local` (`ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

CREATE TABLE IF NOT EXISTS `LTR`.`Arrendatário` (
  `ID` INT NOT NULL,
  `Nome` VARCHAR(45) NULL,
  `DataNascimento` DATE NULL,
  `Gênero` CHAR(1) NULL,
  `ÁreaResidência` VARCHAR(45) NULL,
  `Profissão` TEXT NULL,
  PRIMARY KEY (`ID`));

CREATE TABLE IF NOT EXISTS `LTR`.`Proprietário` (
  `ID` INT NOT NULL,
  `Nome` VARCHAR(45) NULL,
  `DataNascimento` DATE NULL,
  `Gênero` CHAR(1) NULL,
  `ÁreaResidência` VARCHAR(45) NULL,
  `Profissão` TEXT NULL,
  PRIMARY KEY (`ID`));

CREATE TABLE IF NOT EXISTS `LTR`.`GerePropriedade` (
  `Funcionário` INT NULL,
  `Propriedade` INT NULL,
  INDEX `Propriedade_idx` (`Propriedade` ASC) VISIBLE,
  CONSTRAINT `Funcionário`
    FOREIGN KEY (`Funcionário`)
      REFERENCES `LTR`.`Funcionário` (`ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `Propriedade`
    FOREIGN KEY (`Propriedade`)
      REFERENCES `LTR`.`Propriedade` (`ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

CREATE TABLE IF NOT EXISTS `LTR`.`Compra` (
  `ID` INT NOT NULL,
  `Propriedade` INT NULL,
  `AntigoProprietário` INT NULL,
  `NovoProprietário` INT NULL,
  `ValorAquisição` DOUBLE NULL,
  `DataAquisição` DATETIME NULL,
  PRIMARY KEY (`ID`),
  INDEX `NovoProp_idx` (`NovoProprietário` ASC)
  VISIBLE,
  INDEX `Propriedade_idx` (`Propriedade` ASC) VISIBLE,
  INDEX `AntigoProp_idx` (`AntigoProprietário` ASC)
  VISIBLE,
  CONSTRAINT `AntigoProp`
    FOREIGN KEY (`AntigoProprietário`)
      REFERENCES `LTR`.`Proprietário` (`ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `NovoProp`
    FOREIGN KEY (`NovoProprietário`)
      REFERENCES `LTR`.`Proprietário` (`ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `Prop`
    FOREIGN KEY (`Propriedade`)
      REFERENCES `LTR`.`Propriedade` (`ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

CREATE TABLE IF NOT EXISTS `LTR`.`Arrendamento` (
  `ID` INT NOT NULL,
  `Propriedade` INT NULL,
  `Arrendatário` INT NULL,
  `DataAssinatura` DATE NULL,
  `Duração` INT NULL,
  `ValorMensual` DECIMAL(8,2) NULL,
  PRIMARY KEY (`ID`),
  INDEX `Arrendatário_idx` (`Arrendatário` ASC)
  VISIBLE,
  INDEX `Propriedade_idx` (`Propriedade` ASC) VISIBLE,
  CONSTRAINT `Arrendatário`
    FOREIGN KEY (`Arrendatário`)
      REFERENCES `LTR`.`Arrendatário` (`ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `Prop`
    FOREIGN KEY (`Propriedade`)
      REFERENCES `LTR`.`Propriedade` (`ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
```

O mesmo inicializa a base de dados com as tabelas descritas no modelo lógico, com os devidos tipos (VARCHAR / TEXT / INT / DATETIME / DECIMAL)

2. Queries desenvolvidas para dar informações aos utilizadores

Desenvolvemos algumas queries (7 scripts, 18 procedimentos e 2 funções) em SQL para dar alguns dados ao utilizador sobre um certo assunto. Alguns exemplos são:

- Função “getProprietário”: Esta função recebe como argumento um ID de uma propriedade e retorna o proprietário que tem a compra mais recente relativa a essa propriedade;

```
DELIMITER //

CREATE FUNCTION GetProprietário (PropriedadeID INT) RETURNS INT DETERMINISTIC
BEGIN
  SELECT NovoProprietário INTO @return FROM Compra
    WHERE Propriedade = PropriedadeID
    ORDER BY DataAquisição DESC
    LIMIT 1;

  RETURN @return;
END; //

DELIMITER ;

/*
EXEMPLO DE UTILIZAÇÃO: SELECT GetProprietário(<PropriedadeID>) AS Proprietário;
*/
```

- Procedimentos que adicionam e removem um certo elemento de uma certa tabela, por exemplo para remover um proprietário:

```
USE LTR;

DELIMITER //

CREATE PROCEDURE RemoverProprietario(
    IN p_ID INT
)
BEGIN
    DELETE FROM LTR.Compra
    WHERE AntigoProprietário = p_ID OR NovoProprietário = p_ID;

    DELETE FROM LTR.Proprietário
    WHERE ID = p_ID;
END //

DELIMITER ;

/*
EXEMPLO DE UTILIZAÇÃO: CALL RemoverProprietario(<ProprietárioID>);
*/
```

Ao remover um proprietário teremos também de remover todas as Compras a que ele está associado. Logo, o procedimento para remover um proprietário terá de remover as suas entradas na tabela “Compra”, e apenas depois remover o proprietário propriamente dito.

Para adicionar um proprietário, apenas se tem de fazer a inserção na tabela sem ter preocupações com as outras, porém temos de ter cuidado pois o ID que iremos atribuir a esse proprietário terá de ser o último ID, incrementando em 1:

```
USE LTR;

DELIMITER //

CREATE PROCEDURE AdicionarProprietario(
    IN p_Nome VARCHAR(45),
    IN p_DataNascimento DATE,
    IN p_Genero CHAR(1),
    IN p_AreaResidencia VARCHAR(45),
    IN p_Profissao TEXT
)
BEGIN
    DECLARE ultimoID INT;

    SELECT COALESCE(MAX(ID), 0) INTO ultimoID FROM LTR.Proprietário;
    SET ultimoID = ultimoID + 1;

    INSERT INTO LTR.Proprietário (ID, Nome, DataNascimento, Género, ÁreaResidência, Profissão)
    VALUES (ultimoID, p_Nome, p_DataNascimento, p_Genero, p_AreaResidencia, p_Profissao);
END //

DELIMITER ;

/*
EXEMPLO DE UTILIZAÇÃO: CALL AdicionarProprietario('nomeProprietário', 'DataNascimento', 'Género', 'ÁreaResidência', 'Profissão');
*/
```

- Script que mostra a percentagem de riqueza (valor total das propriedades) em relação ao género:

```
SET @ValorTotalM = 0;
SET @ValorTotalF = 0;

SELECT SUM(Valor) INTO @ValorTotalM FROM Propriedade
WHERE (SELECT Género FROM Proprietário WHERE Proprietário.ID = GetProprietário(Propriedade.ID) AND Proprietário.ID != 1) = 'M';

SELECT SUM(Valor) INTO @ValorTotalF FROM Propriedade
WHERE (SELECT Género FROM Proprietário WHERE Proprietário.ID = GetProprietário(Propriedade.ID)) = 'F';

SELECT ROUND(@ValorTotalM / (@ValorTotalM + @ValorTotalF) * 100, 2) AS PercentagemMasculino, ROUND(@ValorTotalF / (@ValorTotalM + @ValorTotalF) * 100, 2) AS PercentagemFeminino;
```

O script começa por declarar 2 variáveis, valor total das propriedades do género masculino e feminino, que serão úteis para fazer as proporções no final do mesmo. Ele

irá colocar a soma dos valores de todas as propriedades cujo funcionário é do gênero masculino na respectiva variável, fazendo depois o mesmo para o gênero feminino.

- Como foi dito no início do ponto 2, desenvolvemos muito mais scripts, procedimentos e funções do que os mostrados aqui, porém por uma questão de simplicidade mostramos apenas alguns!

Conclusão

Neste projeto, desenvolvemos um sistema de gestão de propriedades e arrendamentos chamado LTR, tendo como objetivo proporcionar uma plataforma eficiente e organizada para gerir informações sobre funcionários, proprietários, arrendatários, propriedades, transações de compra e arrendamento.

Durante a análise de requisitos, identificamos os principais elementos necessários para o sistema e organizamos esses requisitos em diferentes categorias, incluindo descrição, manipulação e controle. Essa abordagem ajudou-nos a compreender melhor as necessidades dos utilizadores e a garantir que todas as funcionalidades essenciais fossem consideradas.

O modelo conceptual foi realizado para representar a estrutura geral do sistema, incluindo entidades, relacionamentos e atributos.

Com base no modelo conceptual, procedemos à construção do modelo lógico. Fizemos a normalização dos dados para garantir a integridade e a eficiência do sistema.

Em seguida, implementamos o sistema no ambiente físico, traduzindo o esquema lógico para uma linguagem de consulta estruturada (SQL) compatível com o sistema de gestão de bases de dados escolhido (MYSQL). Desenvolvemos consultas para atender às necessidades requeridas.

Em resumo, o sistema LTR cumpre os objetivos estabelecidos, fornecendo uma solução abrangente para a gestão de propriedades e arrendamentos. Através de uma abordagem cuidadosa de análise, modelação e implementação, criamos um sistema que oferece eficiência, organização e acesso fácil às informações críticas. Esperamos que este projeto seja útil para aqueles envolvidos na gestão de propriedades e arrendamentos, proporcionando uma ferramenta poderosa para facilitar e melhorar os processos envolvidos nessas atividades.

Bibliografia

- <https://dados.gov.pt>
- <https://www.w3schools.com/sql/>
- <https://learn.microsoft.com/en-us/sql/>
- <https://www.youtube.com/watch?v=k3SzhjBqWMY>
- <http://repositorium.sdum.uminho.pt/handle/1822/16>
- <https://www.linkedin.com/pulse/o-que-é-um-modelo-de-dados-lógico-oscar-kenjiro-n-asakura>
- <https://www.guru99.com/database-normalization.html>