

Instituto Politécnico Viana do Castelo

Escola Superior de Tecnologia e Gestão

**Engenharia Informática**

**2018/2019**

17512 Pedro Henrique Veloso da Silva

17485 Diogo Eduardo da Conceição Correia

Relatório de Projeto

da

Unidade Curricular de **Base de Dados**

**Gestão de um Centro de Estudos**

Janeiro de 2019

RESUMO

Este projeto tem como principal objetivo implementar uma base de dados que suporte um sistema de gestão de um centro de Estudos.

Esta base de dados deverá ser capaz de gerir todos os aspetos relacionados com a atividade do centro de Estudos, desde o armazenamento de informações sobre os alunos e professores, até ao registo de pagamentos de mensalidades por parte dos alunos.

Neste relatório apresentamos as varias fases do projeto, desde a fase inicial em que são levantados os requisitos e é elaborado o Diagrama Entidade-Relacionamento até a implementação da base de dados e Gestão dos dados da mesma, com inserção de Dados, atualização de Dados e Consultas na base de dados.

Conteúdo

[1. Introdução 1](#_Toc496611513)

[2. Desenvolvimento do trabalho 2](#_Toc496611514)

[2.1 Levantamento de requisitos 2](#_Toc496611515)

[2.2 Especificação e design 3](#_Toc496611516)

[O modelo de Entidade e Relacionamentos – Versão 1 3](#_Toc496611517)

[O esquema de tabelas 3](#_Toc496611518)

[O esquema de tabelas na 3 FN 4](#_Toc496611519)

[O modelo de Entidade e Relacionamentos – Versão final 6](#_Toc496611520)

[2.3 Implementação da base de dados 7](#_Toc496611521)

[Script de criação de tabelas 7](#_Toc496611522)

[Restrições de integridade 9](#_Toc496611523)

[Script de Criação de vistas 10](#_Toc496611524)

[2.3 Gestão dos dados na base de dados 10](#_Toc496611525)

[Inserção de dados 10](#_Toc496611526)

[Atualização de dados 10](#_Toc496611527)

[Consultas simples 10](#_Toc496611528)

[Consultas com join de tabelas 10](#_Toc496611529)

[Consultas recorrendo ao uso das vistas 10](#_Toc496611530)

[3. Conclusão 11](#_Toc496611531)

[Referências 12](#_Toc496611532)

Índice de figuras

[Fig. 1 - Modelo ER Inicial. 3](#_Toc532402981)

[Fig. 2 - Modelo ER Final. 6](#_Toc532402982)

Índice de Tabelas

**No table of figures entries found.**

# 1. Introdução

O presente relatório insere-se no âmbito da unidade curricular de Base de Dados, com a finalidade de demonstrar os conhecimentos adquiridos nas aulas teóricas e práticas da Unidade Curricular em questão. Sendo assim, ao longo deste relatório serão expostos todos os aspetos inerentes à realização do trabalho prático proposto pelos docentes da Unidade Curricular, sendo o objetivo final do mesmo, a criação e Gestão de uma base de dados que dê suporte à Gestão de um Centro de Estudos, recorrendo ao sql developer e ORACLE Database.

Tratando-se da Gestão de um centro de Estudos, verificamos logo que as principais relações que existem será entre alunos, professores e disciplinas. No entanto, é necessário também ter em conta que existem várias sessões de explicações para cada disciplina, existem pagamentos a ser efetuados pelos alunos, entre outros aspetos de importantes na correta elaboração do trabalho pratico. Para isso é necessário então armazenar vários tipos de dados, desde informações básicas como Nomes, moradas, contactos, até dados mais específicos do tema, como o horário das sessões de explicações, atribuição de professores para as mesmas sessões, valores mensais a pagar pelos alunos, entre outros.

O presente relatório está organizado em 4 vertentes. Na vertente inicial, faz-se um levantamento de requisitos que devem ser cumpridos. A vertente seguinte trata a Especificação e design da base de dados e já na terceira vertente mostra-se como foi feita a sua implementação. Já na última vertente deste relatório, temos vários aspetos relacionados com a Gestão dos dados presentes na referida base de Dados.

No que toca à especificação e design da base de dados, mostraremos todas as fases ultrapassadas na elaboração de um “esboço” daquilo que deveria ser a nossa base de dados. Aqui desenvolvemos o modelo ER(Entidade-Relacionamento) que depois de realizada a devida Normalização, é apresentado na sua forma final, tornando-se muito útil para a seguinte fase, que é a implementação da Base de Dados.

Relativamente à implementação da Base de Dados, apresentamos o script da criação das tabelas, as restrições de integridade e o script de criação de *Views.*

Na base de dados deverá ser possível inserir, atualizar, eliminar e consultar os dados armazenados e é exatamente nisso que a última vertente, a de Gestão dos dados na base de Dados, se foca. Aqui apresentamos scripts de Inserção de Dados, de consultas simples e com joins de tabelas e até recorrendo ao uso das *Views*.

# 2. Desenvolvimento do trabalho

## 2.1 Levantamento de requisitos

A Base de Dados deverá cumprir os seguintes Requisitos:

* Deve ser armazenado para cada Aluno, o seu nome, data de nascimento, morada e o seu contacto.
* Um aluno pode ter explicações de várias disciplinas.
* As disciplinas devem ser identificadas por um código e um nome.
* Uma disciplina pode ser lecionada por vários professores.
* Sobre os Professores deve ser armazenado o seu nome, NIF, data de nascimento, morada e contacto.
* Vários alunos podem participar numa sessão de explicações de uma determinada disciplina.
* As sessões de explicações são lecionadas por apenas um professor.
* Para cada sessão deve ser armazenado além do professor responsável pela mesma e dos alunos que participaram, a data e a hora da sessão.
* Devem ser armazenadas as disciplinas que cada professor está apto a explicar.
* Os alunos efetuam pagamentos mensais pelo que se deve armazenar a data dos pagamentos e o montante.
* O valor dos pagamentos mensais diz respeito ao número de sessões (valor de 2€ por sessão) a que o aluno assistiu no espaço de um mês mais o valor base de 10€.
* Se o aluno não assistir a nenhuma sessão durante o mês, deverá mesmo assim pagar o valor base de 10€.

## 

## 2.2 Especificação e design

## O modelo de Entidade e Relacionamentos – Versão 1

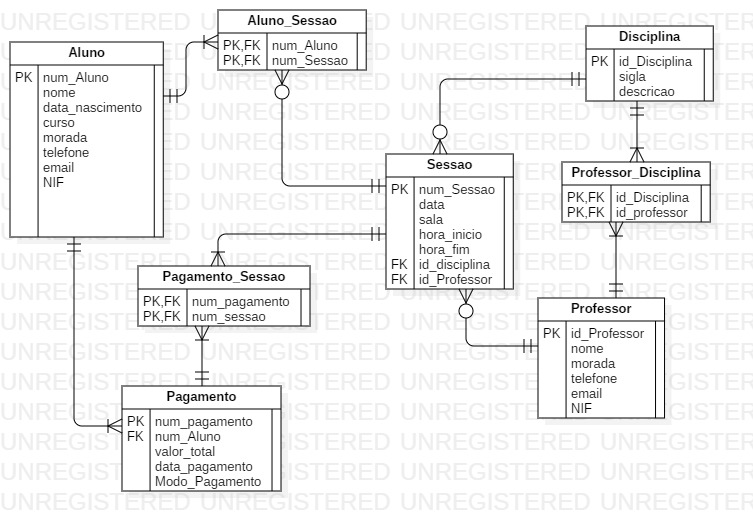


Fig. 1 - Modelo ER Inicial.

### O esquema de tabelas

**ALUNO** (**num\_aluno**, nome, data\_nascimento, curso, morada, telefone, email, NIF)

**PROFESSOR** (**id\_professor**, nome, morada, telefone, email, NIF)

**DISCIPLINA** (**id\_disciplina**, sigla, descricao)

**PROFESSOR\_DISCIPLINA** (**id\_professor, id\_disciplina**)

**SESSAO** (**num\_sessao**, data, sala, hora\_inicio, hora\_fim, id\_disciplina, id\_professor)

**ALUNO\_SESSAO** (**num\_aluno, num\_sessao**)

**PAGAMENTO** (**num\_pagamento**, valor\_total, data\_pagamento, Modo\_Pagamento)

**PAGAMENTO\_SESSAO** (**num\_pagamento, num\_sessao**)

### 

### O esquema de tabelas na 3 FN

A Normalização duma base de dados tem como objetivo principal, a reorganização de um projeto de base de dados de forma a reduzir a redundância de dados, aumentar a integridade de dados, aumentar o desempenho da base de dados e facilitar a manutenção da mesma. A normalização é realizada de modo a que a base de dados se apresente num estado ótimo sem duplicação, nem perda de informação. A Normalização consiste na aplicação de um conjunto de regras que visam verificar o estado em que o modelo relacional se encontra.

Para o nosso projeto vamos verificar se as tabelas estão na 3FN:

1FN

**ALUNO** (**num\_aluno**, nome, data\_nascimento, id\_curso, sigla\_Curso, nome\_curso, tipo\_curso, descricao\_tipo\_curso, rua, nPorta, cPostal, localidade, telefone, email, NIF)

**PROFESSOR** (**id\_professor**, nome, rua, nPorta, cPostal, localidade, telefone, email, NIF)

**DISCIPLINA** (**id\_disciplina**, sigla, descricao)

**PROFESSOR\_DISCIPLINA** (**id\_professor, id\_disciplina**)

**SESSAO** (**num\_sessao**, data, id\_sala, tipo\_sala, descricao\_tipo\_sala, hora\_inicio, hora\_fim, id\_disciplina, id\_professor)

**ALUNO\_SESSAO** (**num\_aluno, num\_sessao**)

**PAGAMENTO** (**num\_pagamento**, valor\_total, data\_pagamento, id\_ModoPag, descricao\_ModoPag)

**PAGAMENTO\_SESSAO** (**num\_pagamento, num\_sessao**)

2FN

**ALUNO** (**num\_aluno**, nome, data\_nascimento, id\_curso, sigla\_Curso, nome\_curso, tipo\_curso, descricao\_tipo\_curso, rua, nPorta, cPostal, localidade, telefone, email, NIF)

**PROFESSOR** (**id\_professor**, nome, rua, nPorta, cPostal, localidade, telefone, email, NIF)

**DISCIPLINA** (**id\_disciplina**, sigla, descricao)

**PROFESSOR\_DISCIPLINA** (**id\_professor, id\_disciplina**)

**SESSAO** (**num\_sessao**, data, id\_sala, tipo\_sala, descricao\_tipo\_sala, hora\_inicio, hora\_fim, id\_disciplina, id\_professor)

**ALUNO\_SESSAO** (**num\_aluno, num\_sessao**)

**PAGAMENTO** (**num\_pagamento**, valor\_total, data\_pagamento, id\_ModoPag, descricao\_ModoPag)

**PAGAMENTO\_SESSAO** (**num\_pagamento, num\_sessao**)

3FN

**TIPOCURSO (id\_tipo,** descricao)

**CURSO (id\_curso,** sigla, nome, id\_tipo)

**CODIGOPOSTAL** (**cPostal**, localidade)

**ALUNO** (**num\_aluno**, nome, data\_nascimento, id\_curso, rua, nPorta, cPostal, telefone, email, NIF)

**PROFESSOR** (**id\_professor**, nome, rua, nPorta, cPostal, telefone, email, NIF)

**DISCIPLINA** (**id\_disciplina**, sigla, descricao)

**PROFESSOR\_DISCIPLINA** (**id\_professor, id\_disciplina**)

**TIPOSALA** (**id\_tipo**, descricao)

**SALA** (**id\_sala**, id\_tipo)

**SESSAO** (**num\_sessao**, data, hora\_inicio, hora\_fim, id\_disciplina, id\_professor, id\_sala)

**ALUNO\_SESSAO** (**num\_aluno, num\_sessao**)

**MODO\_PAGAMENTO** (**id\_ModoPag**, descricao)

**PAGAMENTO** (**num\_pagamento**, valor\_Total, data\_pagamento, num\_aluno, id\_ModoPag)

**PAGAMENTO\_SESSÃO** (**num\_pagamento, num\_sessão**)

Depois de efetuar a Normalização das tabelas, é elaborado um novo modelo relacional (Modelo Relacional Final) e podemos dar inicio à criação da base de dados em Oracle.

### O modelo de Entidade e Relacionamentos – Versão final

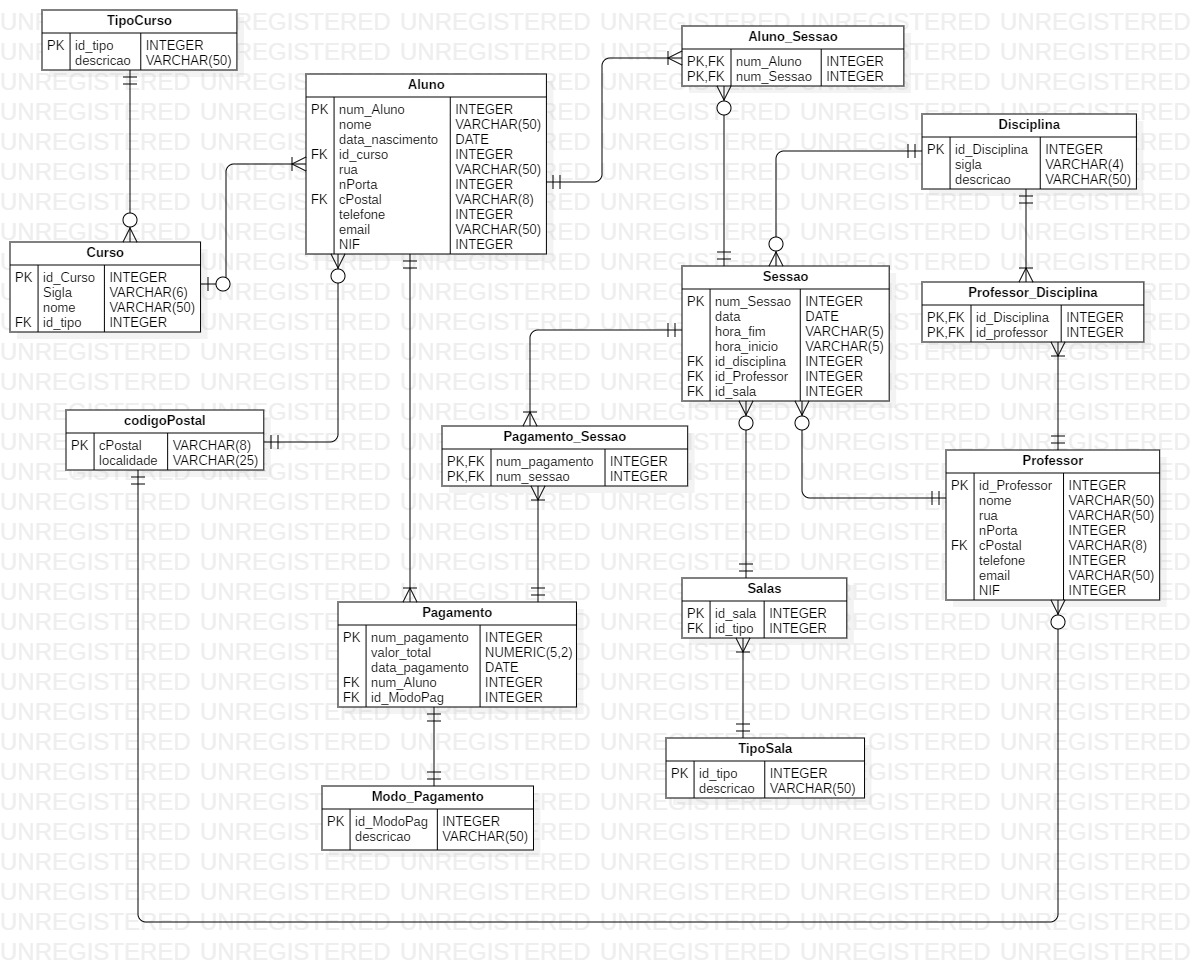


Fig. 2 - Modelo ER Final.

## 

## 2.3 Implementação da base de dados

### Script de criação de tabelas

**CREATE TABLE CODIGOPOSTAL (**

**CPOSTAL VARCHAR(8) NOT NULL,**

**LOCALIDADE VARCHAR(25) NOT NULL,**

**PRIMARY KEY (CPOSTAL)**

**);**

**CREATE TABLE MODO\_PAGAMENTO (**

**ID\_MODOPAG CHAR(4) NOT NULL,**

**DESCRICAO VARCHAR(50) NOT NULL,**

**PRIMARY KEY (ID\_MODOPAG)**

**);**

**CREATE TABLE TIPOSALA (**

**ID\_TIPO INTEGER NOT NULL,**

**DESCRICAO VARCHAR(50) NOT NULL,**

**PRIMARY KEY (ID\_TIPO)**

**);**

**CREATE TABLE SALAS (**

**ID\_SALA CHAR(4) NOT NULL,**

**ID\_TIPO INTEGER NOT NULL,**

**PRIMARY KEY (ID\_SALA)**

**);**

**CREATE TABLE TIPOCURSO (**

**ID\_TIPO INTEGER NOT NULL,**

**DESCRICAO VARCHAR(50) NOT NULL,**

**PRIMARY KEY (ID\_TIPO)**

**);**

**CREATE TABLE CURSO (**

**ID\_CURSO INTEGER NOT NULL,**

**SIGLA VARCHAR(6) NOT NULL,**

**NOME VARCHAR(50) NOT NULL,**

**ID\_TIPO INTEGER NOT NULL,**

**PRIMARY KEY (ID\_CURSO)**

**);**

**CREATE TABLE DISCIPLINA (**

**ID\_DISCIPLINA INTEGER NOT NULL,**

**SIGLA VARCHAR(4) NOT NULL,**

**DESCRICAO VARCHAR(50) NOT NULL,**

**PRIMARY KEY (ID\_DISCIPLINA)**

**);**

**CREATE TABLE PROFESSOR (**

**ID\_PROFESSOR CHAR(4) NOT NULL,**

**NOME VARCHAR(50) NOT NULL,**

**RUA VARCHAR(50) NOT NULL,**

**NPORTA INTEGER NOT NULL,**

**CPOSTAL VARCHAR(8) NOT NULL,**

**TELEFONE INTEGER NOT NULL,**

**EMAIL VARCHAR(50) NOT NULL,**

**NIF INTEGER NOT NULL,**

**PRIMARY KEY (ID\_PROFESSOR)**

**);**

**CREATE TABLE PROFESSOR\_DISCIPLINA (**

**ID\_DISCIPLINA INTEGER NOT NULL,**

**ID\_PROFESSOR CHAR(4) NOT NULL,**

**PRIMARY KEY (ID\_DISCIPLINA, ID\_PROFESSOR)**

**);**

**CREATE TABLE ALUNO (**

**NUM\_ALUNO INTEGER NOT NULL,**

**NOME VARCHAR(50) NOT NULL,**

**DATA\_NASCIMENTO DATE NOT NULL,**

**ID\_CURSO INTEGER NOT NULL,**

**RUA VARCHAR(50) NOT NULL,**

**NPORTA INTEGER NOT NULL,**

**CPOSTAL VARCHAR(8) NOT NULL,**

**TELEFONE INTEGER NOT NULL,**

**EMAIL VARCHAR(50) NOT NULL,**

**NIF INTEGER NOT NULL,**

**PRIMARY KEY (NUM\_ALUNO)**

**);**

**CREATE TABLE SESSAO (**

**NUM\_SESSAO INTEGER NOT NULL,**

**DATA DATE NOT NULL,**

**HORA\_FIM VARCHAR(5) NOT NULL,**

**HORA\_INICIO VARCHAR(5) NOT NULL,**

**ID\_DISCIPLINA INTEGER NOT NULL,**

**ID\_PROFESSOR CHAR(4) NOT NULL,**

**ID\_SALA CHAR(4) NOT NULL,**

**NMENSALIDADE INTEGER NOT NULL,**

**PRIMARY KEY (NUM\_SESSAO)**

**);**

**CREATE TABLE ALUNO\_SESSAO (**

**NUM\_ALUNO INTEGER NOT NULL,**

**NUM\_SESSAO INTEGER NOT NULL,**

**PRIMARY KEY (NUM\_ALUNO, NUM\_SESSAO)**

**);**

**CREATE TABLE PAGAMENTO (**

**NUM\_PAGAMENTO INTEGER NOT NULL,**

**NMENSALIDADE INTEGER NOT NULL,**

**VALOR\_TOTAL NUMERIC(5,2) NOT NULL,**

**DATA\_PAGAMENTO DATE NOT NULL,**

**NUM\_ALUNO INTEGER NOT NULL,**

**ID\_MODOPAG CHAR(4) NOT NULL,**

**PRIMARY KEY (NUM\_PAGAMENTO)**

**);**

**CREATE TABLE MENSALIDADE (**

**NMENSALIDADE INTEGER NOT NULL,**

**MES VARCHAR(20) NOT NULL,**

**ANO VARCHAR(4) NOT NULL,**

**PRIMARY KEY (NMENSALIDADE)**

**);**

**CREATE TABLE ALUNO\_MENSALIDADE (**

**NMENSALIDADE INTEGER NOT NULL,**

**NUM\_ALUNO INTEGER NOT NULL,**

**VALOR NUMERIC(5,2) NOT NULL,**

**PRIMARY KEY (NMENSALIDADE, NUM\_ALUNO)**

**);**

### Restrições de integridade

**ALTER TABLE SALAS ADD FOREIGN KEY (ID\_TIPO) REFERENCES TIPOSALA(ID\_TIPO);**

**ALTER TABLE CURSO ADD FOREIGN KEY (ID\_TIPO) REFERENCES TIPOCURSO(ID\_TIPO);**

**ALTER TABLE PROFESSOR ADD FOREIGN KEY (CPOSTAL) REFERENCES CODIGOPOSTAL(CPOSTAL);**

**ALTER TABLE PROFESSOR\_DISCIPLINA ADD FOREIGN KEY (ID\_DISCIPLINA) REFERENCES DISCIPLINA(ID\_DISCIPLINA);**

**ALTER TABLE PROFESSOR\_DISCIPLINA ADD FOREIGN KEY (ID\_PROFESSOR) REFERENCES PROFESSOR(ID\_PROFESSOR);**

**ALTER TABLE ALUNO ADD FOREIGN KEY (CPOSTAL) REFERENCES CODIGOPOSTAL(CPOSTAL);**

**ALTER TABLE ALUNO ADD FOREIGN KEY (ID\_CURSO) REFERENCES CURSO(ID\_CURSO);**

**ALTER TABLE SESSAO ADD FOREIGN KEY (ID\_SALA) REFERENCES SALAS(ID\_SALA);**

**ALTER TABLE SESSAO ADD FOREIGN KEY (ID\_DISCIPLINA) REFERENCES DISCIPLINA(ID\_DISCIPLINA);**

**ALTER TABLE SESSAO ADD FOREIGN KEY (ID\_PROFESSOR) REFERENCES PROFESSOR(ID\_PROFESSOR);**

**ALTER TABLE SESSAO ADD FOREIGN KEY (NMENSALIDADE) REFERENCES MENSALIDADE(NMENSALIDADE);**

**ALTER TABLE ALUNO\_SESSAO ADD FOREIGN KEY (NUM\_ALUNO) REFERENCES ALUNO(NUM\_ALUNO);**

**ALTER TABLE ALUNO\_SESSAO ADD FOREIGN KEY (NUM\_SESSAO) REFERENCES SESSAO(NUM\_SESSAO);**

**ALTER TABLE PAGAMENTO ADD FOREIGN KEY (ID\_MODOPAG) REFERENCES MODO\_PAGAMENTO(ID\_MODOPAG);**

**ALTER TABLE PAGAMENTO ADD FOREIGN KEY (NUM\_ALUNO) REFERENCES ALUNO(NUM\_ALUNO);**

**ALTER TABLE PAGAMENTO ADD FOREIGN KEY (NMENSALIDADE) REFERENCES MENSALIDADE(NMENSALIDADE);**

**ALTER TABLE ALUNO\_MENSALIDADE ADD FOREIGN KEY (NUM\_ALUNO) REFERENCES ALUNO(NUM\_ALUNO);**

**ALTER TABLE ALUNO\_MENSALIDADE ADD FOREIGN KEY (NMENSALIDADE) REFERENCES MENSALIDADE(NMENSALIDADE);**

### 

### Script de Criação de vistas

* **VIEW PAGAMENTOS**

CREATE VIEW V\_PAGAMENTOS (NRO\_PAGAMENTO,MENSALIDADE, MONTANTE, DATA, NRO\_ALUNO, NOME\_ALUNO, MODO\_PAGAMENTO)

AS

SELECT P.NUM\_PAGAMENTO,P.NMENSALIDADE, P.VALOR\_TOTAL, P.DATA\_PAGAMENTO, P.NUM\_ALUNO, A.NOME, MP.DESCRICAO

FROM PAGAMENTO P, MODO\_PAGAMENTO MP, ALUNO A

WHERE MP.ID\_MODOPAG = P.ID\_MODOPAG AND P.NUM\_ALUNO = A.NUM\_ALUNO

ORDER BY P.NUM\_PAGAMENTO DESC;

* **VIEW ALUNOS**

CREATE VIEW V\_ALUNOS (NRO\_ALUNO, NOME, CURSO, DATA\_NASCIMENTO, RUA, PORTA, CODIGO\_POSTAL, LOCALIDADE, TELEFONE, EMAIL, NIF)

AS

SELECT A.NUM\_ALUNO, A.NOME, C.NOME, A.DATA\_NASCIMENTO, A.RUA, A.NPORTA, A.CPOSTAL, CP.LOCALIDADE, A.TELEFONE, A.EMAIL, A.NIF

FROM ALUNO A, CURSO C, CODIGOPOSTAL CP

WHERE A.ID\_CURSO = C.ID\_CURSO AND A.CPOSTAL = CP.CPOSTAL

ORDER BY NUM\_ALUNO ASC;

* **VIEW PROFESSORES**

CREATE VIEW V\_PROFESSORES (NRO\_PROFESSOR, NOME, RUA, PORTA, CODIGO\_POSTAL, LOCALIDADE, TELEFONE, EMAIL, NIF)

AS

SELECT P.ID\_PROFESSOR, P.NOME, P.RUA, P.NPORTA, P.CPOSTAL, CP.LOCALIDADE, P.TELEFONE, P.EMAIL, P.NIF

FROM PROFESSOR P, CODIGOPOSTAL CP

WHERE P.CPOSTAL = CP.CPOSTAL

ORDER BY ID\_PROFESSOR ASC;

* **VIEW DISCIPLINAS**

CREATE VIEW V\_DISCIPLINAS (ID\_DISCIPLINA, SIGLA, NOME)

AS

SELECT ID\_DISCIPLINA, SIGLA, DESCRICAO FROM DISCIPLINA

ORDER BY ID\_DISCIPLINA;

* **VIEW SESSÕES**

CREATE VIEW V\_SESSOES (NRO\_SESSAO, DATA, HORA\_INICIO, HORA\_FIM, DISCIPLINA, PROFESSOR, SALA)

AS

SELECT S.NUM\_SESSAO, S.DATA, S.HORA\_INICIO, S.HORA\_FIM, D.SIGLA, P.NOME, S.ID\_SALA

FROM SESSAO S, DISCIPLINA D, PROFESSOR P

WHERE S.ID\_DISCIPLINA = D.ID\_DISCIPLINA AND S.ID\_PROFESSOR = P.ID\_PROFESSOR

ORDER BY S.NUM\_SESSAO DESC;

## 2.4 Gestão dos dados na base de dados

A base de dados pode ser usada para armazenar a informação por um longo período de tempo, para isso deverá ser possível inserir, atualizar, eliminar e consultar os dados armazenados.

…..

### Inserção de dados

* **INSERT CODIGOS POSTAIS**

INSERT INTO CODIGOPOSTAL(CPOSTAL, LOCALIDADE) VALUES('4970-555','SÃO COSME E SÃO DAMIÃO');

INSERT INTO CODIGOPOSTAL(CPOSTAL, LOCALIDADE) VALUES('4970-441','PRAÇA MUNICIPAL');

INSERT INTO CODIGOPOSTAL(CPOSTAL, LOCALIDADE) VALUES('4900-357','RUA SANTIAGO DA BARRA');

* **INSERT TIPOS DE CURSO**

INSERT INTO TIPOCURSO(ID\_TIPO, DESCRICAO) VALUES(1,'CURSO PROFISSIONAL');

INSERT INTO TIPOCURSO(ID\_TIPO, DESCRICAO) VALUES(2,'LICENCIATURA');

INSERT INTO TIPOCURSO(ID\_TIPO, DESCRICAO) VALUES(3,'MESTRADO');

INSERT INTO TIPOCURSO(ID\_TIPO, DESCRICAO) VALUES(4,'DOUTORAMENTO');

* **INSERT CURSOS**

INSERT INTO CURSO(ID\_CURSO, SIGLA, NOME, ID\_TIPO) VALUES(9119, 'EI', 'ENGENHARIA INFORMATICA', 2);

INSERT INTO CURSO(ID\_CURSO, SIGLA, NOME, ID\_TIPO) VALUES(9147,'G','GESTÃO',2);

INSERT INTO CURSO(ID\_CURSO, SIGLA, NOME, ID\_TIPO) VALUES(6797,'MTGSI','MESTRADO TEC. E GESTÃO SISTEMAS DE INFORMAÇÃO',3);

INSERT INTO CURSO(ID\_CURSO, SIGLA, NOME, ID\_TIPO) VALUES(067,'CTSPIG','CTSP DE INFORMATICA DE GESTÃO',1);

INSERT INTO CURSO(ID\_CURSO, SIGLA, NOME, ID\_TIPO) VALUES(5141,'DEI','DOUTORAMENTO ENGENHARIA INFORMATICA',4);

* **INSERT TIPOSALA**

INSERT INTO TIPOSALA(ID\_TIPO, DESCRICAO) VALUES(1,'SALA');

INSERT INTO TIPOSALA(ID\_TIPO, DESCRICAO) VALUES(2,'LABORATORIO');

INSERT INTO TIPOSALA(ID\_TIPO, DESCRICAO) VALUES(3,'AUDITORIO');

* **INSERT SALAS**

INSERT INTO SALAS(ID\_SALA, ID\_TIPO) VALUES('S1.1',1);

INSERT INTO SALAS(ID\_SALA, ID\_TIPO) VALUES('S1.2',1);

INSERT INTO SALAS(ID\_SALA, ID\_TIPO) VALUES('S2.1',1);

INSERT INTO SALAS(ID\_SALA, ID\_TIPO) VALUES('S2.2',1);

INSERT INTO SALAS(ID\_SALA, ID\_TIPO) VALUES('S2.3',1);

INSERT INTO SALAS(ID\_SALA, ID\_TIPO) VALUES('L1.1',2);

INSERT INTO SALAS(ID\_SALA, ID\_TIPO) VALUES('L1.2',2);

INSERT INTO SALAS(ID\_SALA, ID\_TIPO) VALUES('L1.3',2);

INSERT INTO SALAS(ID\_SALA, ID\_TIPO) VALUES('A1.1',3);

INSERT INTO SALAS(ID\_SALA, ID\_TIPO) VALUES('A2.1',3);

* **INSERT DISCIPLINAS**

INSERT INTO DISCIPLINA(ID\_DISCIPLINA, SIGLA, DESCRICAO) VALUES(1,'BD','BASE DE DADOS');

INSERT INTO DISCIPLINA(ID\_DISCIPLINA, SIGLA, DESCRICAO) VALUES(2,'ASC','ARQUITETURA E SISTEMAS DE COMPUTADORES');

INSERT INTO DISCIPLINA(ID\_DISCIPLINA, SIGLA, DESCRICAO) VALUES(3,'IHM','INTERAÇÃO HOMEM-MAQUINA');

INSERT INTO DISCIPLINA(ID\_DISCIPLINA, SIGLA, DESCRICAO) VALUES(4,'IA','INTELIGENCIA ARTIFICIAL');

INSERT INTO DISCIPLINA(ID\_DISCIPLINA, SIGLA, DESCRICAO) VALUES(5,'ABD','ADMINISTRAÇÃO DE BASE DE DADOS');

* **INSERT PROFESSORES**

INSERT INTO PROFESSOR(ID\_PROFESSOR, NOME, RUA, NPORTA, CPOSTAL, TELEFONE, EMAIL, NIF)

VALUES ('P001','VASCO SILVA','RUA SANTIAGO DA BARRA',17,'4900-357', 965789987, 'vascosilva@ipvc.pt', 123456789);

INSERT INTO PROFESSOR(ID\_PROFESSOR, NOME, RUA, NPORTA, CPOSTAL, TELEFONE, EMAIL, NIF)

VALUES('P002', 'ANA MIRANDA', 'CASTANHEIRA', 54, '4970-555', 912343211, 'mirandaana@ipvc.pt', 987654321);

INSERT INTO PROFESSOR(ID\_PROFESSOR, NOME, RUA, NPORTA, CPOSTAL, TELEFONE, EMAIL, NIF)

VALUES('P003','ANTONIO ALVES','RUA DA LIBERDADE', 7, '4970-441', 933444555, 'a.alves@ipvc.pt', 543219876);

* **INSERT PROFESSOR\_DISCIPLINA**

INSERT INTO PROFESSOR\_DISCIPLINA(ID\_DISCIPLINA, ID\_PROFESSOR) VALUES(1,'P003');

INSERT INTO PROFESSOR\_DISCIPLINA(ID\_DISCIPLINA, ID\_PROFESSOR) VALUES(2,'P001');

INSERT INTO PROFESSOR\_DISCIPLINA(ID\_DISCIPLINA, ID\_PROFESSOR) VALUES(3,'P002');

INSERT INTO PROFESSOR\_DISCIPLINA(ID\_DISCIPLINA, ID\_PROFESSOR) VALUES(4,'P002');

INSERT INTO PROFESSOR\_DISCIPLINA(ID\_DISCIPLINA, ID\_PROFESSOR) VALUES(5,'P001');

* **INSERT ALUNO**

INSERT INTO ALUNO(NUM\_ALUNO, NOME, DATA\_NASCIMENTO, ID\_CURSO, RUA, NPORTA, CPOSTAL, TELEFONE, EMAIL, NIF)

VALUES(17512,'PEDRO SILVA',TO\_DATE('18/03/1996', 'DD/MM/YYYY'), 9119, 'GEREI',105,'4970-555',933333333,'phvsilva@ipvc.pt',123456789);

INSERT INTO ALUNO(NUM\_ALUNO, NOME, DATA\_NASCIMENTO, ID\_CURSO, RUA, NPORTA, CPOSTAL, TELEFONE, EMAIL, NIF)

VALUES(17485,'DIOGO CORREIA',TO\_DATE('21/07/1998', 'DD/MM/YYYY'), 5141, 'ZINDE',10,'4970-441',933543768,'diogo@ipvc.pt',635643587);

INSERT INTO ALUNO(NUM\_ALUNO, NOME, DATA\_NASCIMENTO, ID\_CURSO, RUA, NPORTA, CPOSTAL, TELEFONE, EMAIL, NIF)

VALUES(6875,'RITA BRANCO',TO\_DATE('18/05/1987', 'DD/MM/YYYY'), 9147, 'AVENIDA FIXE',132,'4970-555',961345765,'rita@ipvc.pt',654321987);

INSERT INTO ALUNO(NUM\_ALUNO, NOME, DATA\_NASCIMENTO, ID\_CURSO, RUA, NPORTA, CPOSTAL, TELEFONE, EMAIL, NIF)

VALUES(342,'PAULO LOPES',TO\_DATE('02/01/1986', 'DD/MM/YYYY'), 67, 'RUA DA CIDADE',46,'4900-357',933456743,'paulo@ipvc.pt',456789321);

* **INSERT MENSALIDADE**

INSERT INTO MENSALIDADE VALUES(NMENSALIDADE.NEXTVAL, TO\_CHAR(EXTRACT(MONTH FROM SYSDATE)), TO\_CHAR(EXTRACT(YEAR FROM SYSDATE)));

INSERT INTO MENSALIDADE VALUES(NMENSALIDADE.NEXTVAL,'01','2019');

INSERT INTO MENSALIDADE VALUES(NMENSALIDADE.NEXTVAL, '02','2019');

INSERT INTO MENSALIDADE VALUES(NMENSALIDADE.NEXTVAL, '03','2019');

INSERT INTO MENSALIDADE VALUES(NMENSALIDADE.NEXTVAL, '04','2019');

INSERT INTO MENSALIDADE VALUES(NMENSALIDADE.NEXTVAL, '05','2019');

* **INSERT SESSAO**

INSERT INTO SESSAO(NUM\_SESSAO, DATA, HORA\_INICIO, HORA\_FIM, ID\_DISCIPLINA, ID\_PROFESSOR, ID\_SALA, NMENSALIDADE) VALUES(NUM\_SESSAO.NEXTVAL, TO\_DATE('13/12/2018','DD/MM/YYYY'), '10H00', '12H00',2,'P001','L1.1',1);

INSERT INTO SESSAO(NUM\_SESSAO, DATA, HORA\_INICIO, HORA\_FIM, ID\_DISCIPLINA, ID\_PROFESSOR, ID\_SALA, NMENSALIDADE) VALUES(NUM\_SESSAO.NEXTVAL,SYSDATE,'14H00','16H00',3,'P002','S2.1',1);

INSERT INTO SESSAO(NUM\_SESSAO, DATA, HORA\_INICIO, HORA\_FIM, ID\_DISCIPLINA, ID\_PROFESSOR, ID\_SALA, NMENSALIDADE) VALUES(NUM\_SESSAO.NEXTVAL, TO\_DATE('02/12/2018','DD/MM/YYYY'), '10H00','12H00',5,'P003','L1.2',1);

* **INSERT ALUNO\_MENSALIDADE**

INSERT INTO ALUNO\_MENSALIDADE(NUM\_ALUNO, NMENSALIDADE, VALOR) VALUES(17512, 1, 10);

INSERT INTO ALUNO\_MENSALIDADE(NUM\_ALUNO, NMENSALIDADE, VALOR) VALUES(17485, 1, 10);

INSERT INTO ALUNO\_MENSALIDADE(NUM\_ALUNO, NMENSALIDADE, VALOR) VALUES(6875, 1, 10);

INSERT INTO ALUNO\_MENSALIDADE(NUM\_ALUNO, NMENSALIDADE, VALOR) VALUES(342, 1, 10);

INSERT INTO ALUNO\_MENSALIDADE(NUM\_ALUNO, NMENSALIDADE, VALOR) VALUES(17512, 2, 10);

INSERT INTO ALUNO\_MENSALIDADE(NUM\_ALUNO, NMENSALIDADE, VALOR) VALUES(17485, 2, 10);

INSERT INTO ALUNO\_MENSALIDADE(NUM\_ALUNO, NMENSALIDADE, VALOR) VALUES(6875, 2, 10);

INSERT INTO ALUNO\_MENSALIDADE(NUM\_ALUNO, NMENSALIDADE, VALOR) VALUES(342, 2, 10);

* **INSERT ALUNO\_SESSAO**

INSERT INTO ALUNO\_SESSAO(NUM\_ALUNO, NUM\_SESSAO) VALUES(17512, 1);

INSERT INTO ALUNO\_SESSAO(NUM\_ALUNO, NUM\_SESSAO) VALUES(17512, 2);

INSERT INTO ALUNO\_SESSAO(NUM\_ALUNO, NUM\_SESSAO) VALUES(17485, 1);

INSERT INTO ALUNO\_SESSAO(NUM\_ALUNO, NUM\_SESSAO) VALUES(17485, 2);

INSERT INTO ALUNO\_SESSAO(NUM\_ALUNO, NUM\_SESSAO) VALUES(6875, 1);

INSERT INTO ALUNO\_SESSAO(NUM\_ALUNO, NUM\_SESSAO) VALUES(342, 1);

INSERT INTO ALUNO\_SESSAO(NUM\_ALUNO, NUM\_SESSAO) VALUES(342, 2);

INSERT INTO ALUNO\_SESSAO(NUM\_ALUNO, NUM\_SESSAO) VALUES(342, 4);

INSERT INTO ALUNO\_SESSAO(NUM\_ALUNO, NUM\_SESSAO) VALUES(17485, 3);

INSERT INTO ALUNO\_SESSAO(NUM\_ALUNO, NUM\_SESSAO) VALUES(17485, 4);

INSERT INTO ALUNO\_SESSAO(NUM\_ALUNO, NUM\_SESSAO) VALUES(6875, 4);

* **INSERT MODO\_PAGAMENTO**

INSERT INTO MODO\_PAGAMENTO(ID\_MODOPAG, DESCRICAO) VALUES('MB','MULTIBANCO');

INSERT INTO MODO\_PAGAMENTO(ID\_MODOPAG, DESCRICAO) VALUES('D','DINHEIRO');

INSERT INTO MODO\_PAGAMENTO(ID\_MODOPAG, DESCRICAO) VALUES('C','CHEQUE');

* **INSERT PAGAMENTO**

INSERT INTO PAGAMENTO(NUM\_PAGAMENTO,NMENSALIDADE, VALOR\_TOTAL, DATA\_PAGAMENTO, NUM\_ALUNO, ID\_MODOPAG) VALUES(NUM\_PAGAMENTO.NEXTVAL,1, 0, SYSDATE, 17512, 'MB');

INSERT INTO PAGAMENTO(NUM\_PAGAMENTO,NMENSALIDADE, VALOR\_TOTAL, DATA\_PAGAMENTO, NUM\_ALUNO, ID\_MODOPAG) VALUES(NUM\_PAGAMENTO.NEXTVAL,1, 30, SYSDATE, 17485, 'MB');

INSERT INTO PAGAMENTO(NUM\_PAGAMENTO,NMENSALIDADE, VALOR\_TOTAL, DATA\_PAGAMENTO, NUM\_ALUNO, ID\_MODOPAG) VALUES(NUM\_PAGAMENTO.NEXTVAL,1, 24, SYSDATE, 342, 'MB');

INSERT INTO PAGAMENTO(NUM\_PAGAMENTO,NMENSALIDADE, VALOR\_TOTAL, DATA\_PAGAMENTO, NUM\_ALUNO, ID\_MODOPAG) VALUES(NUM\_PAGAMENTO.NEXTVAL,1, 14, SYSDATE, 6875, 'D');

INSERT INTO PAGAMENTO(NUM\_PAGAMENTO,NMENSALIDADE, VALOR\_TOTAL, DATA\_PAGAMENTO, NUM\_ALUNO, ID\_MODOPAG) VALUES(NUM\_PAGAMENTO.NEXTVAL,2, 100, SYSDATE, 6875, 'D');

INSERT INTO PAGAMENTO(NUM\_PAGAMENTO,NMENSALIDADE, VALOR\_TOTAL, DATA\_PAGAMENTO, NUM\_ALUNO, ID\_MODOPAG) VALUES(NUM\_PAGAMENTO.NEXTVAL,2, 14, SYSDATE, 17512, 'D');

* **CRIAÇÃO DE SEQUENCIAS**

CREATE SEQUENCE NUM\_SESSAO

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE NMENSALIDADE

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE NUM\_PAGAMENTO

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

### Atualização de dados

* **TRIGGER PARA ATUALIZAÇAO DO VALOR DA MENSALIDADE DEPOIS DE INSERIR AS SESSOES A QUE O ALUNO ASSISTIU**

CREATE OR REPLACE TRIGGER ADDSESSAOMENSALIDADE

AFTER UPDATE OR INSERT OR DELETE ON ALUNO\_SESSAO

FOR EACH ROW

DECLARE NROMENSALIDADE INTEGER;

BEGIN

SELECT SESSAO.NMENSALIDADE INTO NROMENSALIDADE

FROM SESSAO

WHERE :NEW.NUM\_SESSAO = SESSAO.NUM\_SESSAO;

UPDATE ALUNO\_MENSALIDADE SET ALUNO\_MENSALIDADE.VALOR = ALUNO\_MENSALIDADE.VALOR + 2 WHERE ALUNO\_MENSALIDADE.NUM\_ALUNO = :NEW.NUM\_ALUNO

AND ALUNO\_MENSALIDADE.NMENSALIDADE = NROMENSALIDADE;

END;

* **TRIGGER PARA DEFINIR AUTOMATICAMENTE O VALOR TOTAL DO PAGAMENTO COMO SENDO O VALOR DA MENSALIDADE DO ALUNO INSERIDOS**

CREATE OR REPLACE TRIGGER ADDVALORPAGAMENTO

BEFORE INSERT ON PAGAMENTO

FOR EACH ROW

DECLARE VALOR FLOAT;

BEGIN

SELECT ALUNO\_MENSALIDADE.VALOR INTO VALOR

FROM ALUNO\_MENSALIDADE

WHERE ALUNO\_MENSALIDADE.NMENSALIDADE = :NEW.NMENSALIDADE

AND ALUNO\_MENSALIDADE.NUM\_ALUNO = :NEW.NUM\_ALUNO;

:NEW.VALOR\_TOTAL := VALOR;

END;

### Consultas simples

* **LISTAR TODOS OS ALUNOS:**

SELECT \* FROM ALUNO;

* **LISTAR TODOS OS PROFESSORES:**

SELECT \* FROM PROFESSOR;

* **LISTAR ID, NOME E CONTACTO DOS PROFESSORES:**

SELECT ID\_PROFESSOR, NOME, TELEFONE FROM PROFESSOR;

* **LISTAR TODAS AS SESSÕES DE EXPLICAÇÕES**:

SELECT \* FROM SESSAO;

* **VER O VALOR TOTAL DOS PAGAMENTOS:**

SELECT SUM(VALOR\_TOTAL) AS RENDIMENTO FROM PAGAMENTO;

* **VER O NUMERO DE SESSÕES REGISTADAS NA BASE DE DADOS:**

SELECT COUNT(\*) AS TOTAL\_SESSOES FROM SESSAO;

* **VER O NUMERO TOTAL DE ALUNOS REGISTADOS NA BASE DE DADOS:**

SELECT COUNT(\*) AS TOTAL\_ALUNOS FROM ALUNO;

* **VER O NUMERO TOTAL DE SALAS REGISTADAS NA BASE DE DADOS:**

SELECT COUNT(\*) AS NRO\_DE\_SALAS FROM SALAS;

* **VER O NUMERO TOTAL DE CURSOS REGISTADOS NA BASE DE DADOS:**

SELECT COUNT(\*) AS TOTAL\_CURSOS FROM CURSO;

* **LISTAR TODAS AS SESSÕES DE 2018:**

SELECT \* FROM SESSAO WHERE DATA >= TO\_DATE('01.01.2018','DD.MM.YYYY') AND DATA < TO\_DATE('01.01.2019','DD.MM.YYYY');

### Consultas com **join** de tabelas

* **LISTAR O ID, NOME E CONTACTO DOS ALUNOS QUE ASSISTIRAM À SESSAO 4:**

SELECT A.NUM\_ALUNO, A.NOME, A.TELEFONE

FROM ALUNO A, ALUNO\_SESSAO A\_S

WHERE A.NUM\_ALUNO = A\_S.NUM\_ALUNO AND A\_S.NUM\_SESSAO=4;

* **LISTAR PAGAMENTOS PARA O ALUNO “PEDRO SILVA”:**

SELECT P.NUM\_PAGAMENTO, P.NUM\_ALUNO, A.NOME, P.DATA\_PAGAMENTO, P.VALOR\_TOTAL

FROM PAGAMENTO P, ALUNO A

WHERE P.NUM\_ALUNO = A.NUM\_ALUNO AND A.NOME LIKE 'PEDRO SILVA';

* **VER QUAIS AS DISCIPLINAS LECIONADAS NAS SESSOES REGISTADAS:**

SELECT S.NUM\_SESSAO, D.SIGLA, D.DESCRICAO

FROM SESSAO S, DISCIPLINA D

WHERE S.ID\_DISCIPLINA = D.ID\_DISCIPLINA;

### Consultas recorrendo ao uso das vistas

# 3. Conclusão

…….

# Referências

1. …
2. …
3. …