



INSTITUTO POLITÉCNICO
DE VIANA DO CASTELO

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
2. Desenvolvimento do trabalho	2
2.1 Levantamento de requisitos	2
2.2 Especificação e design.....	3
O modelo de Entidade e Relacionamentos – Versão 1.....	3
O esquema de tabelas	3
O esquema de tabelas na 3 FN	4
O modelo de Entidade e Relacionamentos – Versão final	6
O modelo de Entidade e Relacionamentos – Versão Final (Oracle).....	6
2.3 Implementação da base de dados	7
Script de criação de tabelas.....	7
Restrições de integridade	10
Script de Criação de vistas	11
Resultado da Criação das Vistas	12
2.4 Gestão dos dados na base de dados	13
Inserção de dados.....	13
Atualização de dados.....	18
Consultas simples	19
Consultas com join de tabelas.....	20
Consultas recorrendo ao uso das vistas	20
3. Conclusão	21
Referências.....	22

ÍNDICE DE FIGURAS

FIG. 1 - MODELO ER INICIAL.....	3
FIG. 2 - MODELO ER FINAL.....	6
FIG. 3 - MODELO ER ORACLE.....	6
FIG. 4 - VIEW V_ALUNOS.....	12
FIG. 5 - VIEW V_DISCIPLINAS.....	12
FIG. 6 - VIEW V_PAGAMENTOS.....	12
FIG. 7 - VIEW V_PROFESSORES.....	12
FIG. 8 - VIEW V_SESSOES.....	12

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 prático. 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

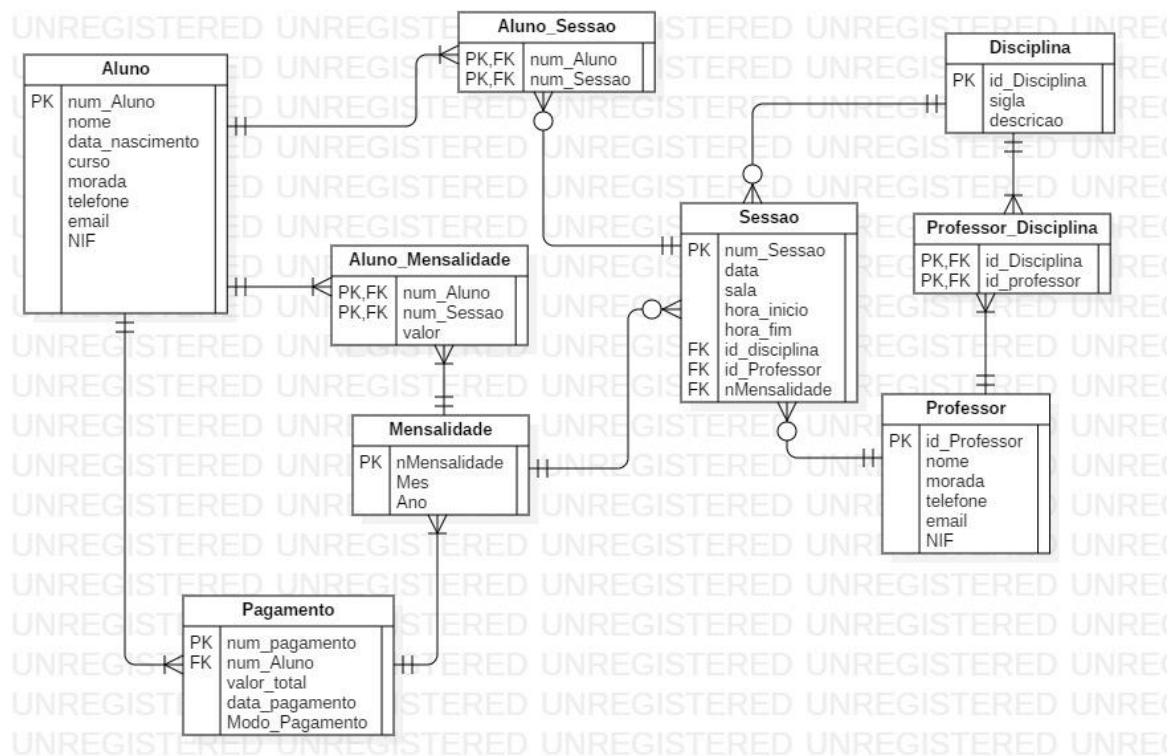


Fig. 1 - Modelo ER Inicial.

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, nMensalidade)

ALUNO_SESSAO (num_aluno, num_sessao)

PAGAMENTO (num_pagamento, valor_total, data_pagamento, Modo_Pagamento)

MENSALIDADE (nMensalidade, Mes, Ano)

ALUNO_MENSALIDADE (num_aluno, nMensalidade)

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, nMensalidade)

ALUNO_SESSAO (num_aluno, num_sessao)

PAGAMENTO (num_pagamento, valor_total, data_pagamento, id_ModoPag, descricao_ModoPag)

PAGAMENTO_SESSAO (num_pagamento, num_sessao)

MENSALIDADE(nMensalidade, Mes, Ano)

ALUNO_MENSALIDADE(num_aluno, nMensalidade)

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, nMensalidade)

ALUNO_SESSAO (num_aluno, num_sessao)

PAGAMENTO (num_pagamento, valor_total, data_pagamento, id_ModoPag, descricao_ModoPag)

PAGAMENTO_SESSAO (num_pagamento, num_sessao)

MENSALIDADE(nMensalidade, Mes, Ano)

ALUNO_MENSALIDADE(num_aluno, nMensalidade)

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, nMensalidade)

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)

MENSALIDADE(nMensalidade, Mes, Ano)

ALUNO_MENSALIDADE(num_aluno, nMensalidade)

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.

[illegible]

```

    graph TD
        PC[PEDRO_CURSO  
P: * ID_CURSO NUMBER (7,5)  
  * SIGLA VARCHAR2 (5 BYTE)  
  * NOME VARCHAR2 (50 BYTE)  
  F: ID_TIPO NUMBER (7,5)  
  PK CURSO_ID, ID_CURSO  
  SYS_C00072 (ID_CURSO)  
  SYS_C00073 (ID_TIPO)]
        PA[PEDRO_ALUNO  
P: * NUM_ALUNO NUMBER (7,5)  
  * NOME VARCHAR2 (50 BYTE)  
  * DATA_PAGAMENTO DATE  
  * ID_CURSO NUMBER (7,5)  
  * RUA VARCHAR2 (50 BYTE)  
  * INPORTA NUMBER (7,5)  
  * CPDSTAL VARCHAR2 (50 BYTE)  
  * TELEFONE VARCHAR2 (50 BYTE)  
  * EMAIL VARCHAR2 (50 BYTE)  
  * NF NUMBER (7,5)  
  PK ALUNO_ID, NUM_ALUNO  
  SYS_C00074 (CPDSTAL)  
  SYS_C00077 (ID_CURSO)]
        PS[PEDRO_SESSAO  
P: * NUM_SESSAO NUMBER (7,5)  
  * DATA DATE  
  * HORA_FIM VARCHAR2 (50 BYTE)  
  * HORA_INCO VARCHAR2 (50 BYTE)  
  F: ID_DISCIPLINA NUMBER (7,5)  
  F: ID_PROFESSOR CHARV (4 BYTE)  
  F: ID_SALA CHARV (4 BYTE)  
  F: INERIALIDADE NUMBER (7,5)  
  PK SESSAO_ID, NUM_SESSAO  
  SYS_C00078 (ID_SALA)  
  SYS_C00079 (ID_DISCIPLINA)  
  SYS_C00080 (ID_PROFESSOR)  
  SYS_C00081 (INERIALIDADE)]
        PD[PEDRO_DISCIPLINA  
P: * ID_DISCIPLINA NUMBER (7,5)  
  * SIGLA VARCHAR2 (50 BYTE)  
  * DESCRCAO VARCHAR2 (50 BYTE)  
  PK DISCIPLINA_ID, ID_DISCIPLINA]
        PP[PEDRO_PROFESSOR  
P: * ID_PROFESSOR CHARV (4 BYTE)  
  * NOME VARCHAR2 (50 BYTE)  
  * RUA VARCHAR2 (50 BYTE)  
  * INPORTA NUMBER (7,5)  
  * CPDSTAL VARCHAR2 (50 BYTE)  
  * TELEFONE VARCHAR2 (50 BYTE)  
  * EMAIL VARCHAR2 (50 BYTE)  
  * NF NUMBER (7,5)  
  PK PROFESSOR_ID, ID_PROFESSOR  
  SYS_C00082 (CPDSTAL)]
        PMP[PEDRO_MODALIDADE  
P: * INERIALIDADE NUMBER (7,5)  
  * VALOR NUMBER (5,2)  
  PK ALUNO_MODALIDADE_ID, NUM_ALUNO  
  SYS_C00083 (NUM_ALUNO)  
  SYS_C00084 (INERIALIDADE)]
        PMS[PEDRO_MODALIDADE_SESSAO  
P: * INERIALIDADE NUMBER (7,5)  
  * VALOR VARCHAR2 (50 BYTE)  
  * MES VARCHAR2 (4 BYTE)  
  * ANO VARCHAR2 (4 BYTE)  
  PK MODALIDADE_ID, INERIALIDADE]
        PPD[PEDRO_PAGAMENTO  
P: * NUM_PAGAMENTO NUMBER (7,5)  
  * INERIALIDADE NUMBER (7,5)  
  * VALOR_TOTAL NUMBER (5,2)  
  * DATA_PAGAMENTO DATE  
  * NUM_ALUNO NUMBER (7,5)  
  * ID_MODALIDAD CHARV (4 BYTE)  
  PK PAGAMENTO_ID, NUM_PAGAMENTO  
  SYS_C00085 (ID_MODALIDAD)  
  SYS_C00086 (NUM_ALUNO)  
  SYS_C00087 (INERIALIDADE)]
        PPS[PEDRO_PAGAMENTO_SALAS  
P: * ID_MODALIDAD CHARV (4 BYTE)  
  * DESCRCAO VARCHAR2 (50 BYTE)  
  PK MODO_PAGAMENTO_ID, ID_MODALIDAD]
        PPSA[PEDRO_TIPOSALA  
P: * ID_TIPO NUMBER (7,5)  
  * DESCRCAO VARCHAR2 (50 BYTE)  
  PK TIPOSALA_ID, ID_TIPO]
        PPSA --> PS
        PPSA --> PP
        PPSA --> PMS
        PPSA --> PPD
        PPSA --> PPS
        PPSA --> PPSA

        PC --> PA
        PC --> PS
        PC --> PD
        PC --> PP
        PC --> PMP
        PC --> PMS
        PC --> PPD
        PC --> PPS
        PC --> PPSA

        PA --> PS
        PA --> PD
        PA --> PP
        PA --> PMP
        PA --> PMS
        PA --> PPD
        PA --> PPS
        PA --> PPSA

        PS --> PD
        PS --> PP
        PS --> PMS
        PS --> PPD
        PS --> PPS
        PS --> PPSA

        PD --> PP
        PD --> PMS
        PD --> PPD
        PD --> PPS
        PD --> PPSA

        PP --> PMS
        PP --> PPD
        PP --> PPS
        PP --> PPSA

        PMP --> PPD
        PMP --> PPS
        PMP --> PPSA

        PMS --> PPD
        PMS --> PPS
        PMS --> PPSA

        PPD --> PPS
        PPD --> PPSA

        PPS --> PPSA
    
```

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;
```

RESULTADO DA CRIAÇÃO DAS VISTAS

NRO_ALUNO	NOME	CURSO	DATA_NASCIMENTO	RUA	PORTA	CODIGO_POSTAL	LOCALIDADE	TELEFONE	EMAIL	NIF
1	342 PAULO LOPES	CTSP DE INFORMATICA...	86.01.02	RUA DA CIDADE		46 4900-357	RUA SANTIAGO DA BARRA	933456743	paulo@ipvc.pt	456789321
2	6875 RITA BRANCO	GESTÃO	87.05.18	AVENIDA FIXE		132 4970-555	SÃO COSME E SÃO DAMIÃO	961345765	rita@ipvc.pt	654321987
3	17485 DIOGO CORREIA	DOCTORAMENTO ENGEN...	98.07.21	ZINDE		10 4970-441	PRAÇA MUNICIPAL	933543768	diogo@ipvc.pt	635643587
4	17512 PEDRO SILVA	ENGENHARIA INFORMA...	96.03.18	GEREI		105 4970-555	SÃO COSME E SÃO DAMIÃO	933333333	phvsilva@ipvc.pt	123456789

Fig. 4 - View V_ALUNOS

ID_DISCIPLINA	SIGLA	NOME
1	BD	BASE DE DADOS
2	ASC	ARQUITETURA E SISTEMAS DE COMPUTADORES
3	IHM	INTERAÇÃO HOMEM-MAQUINA
4	IA	INTELIGENCIA ARTIFICIAL
5	ABD	ADMINISTRAÇÃO DE BASE DE DADOS

Fig. 5 - View V_DISCIPLINAS

NRO_PAGAMENTO	MENSALIDADE	MONTANTE	DATA	NRO_ALUNO	NOME_ALUNO	MODO_PAGAMENTO
1	6	2	10 19.01.02	17512	PEDRO SILVA	DINHEIRO
2	5	2	12 19.01.02	6875	RITA BRANCO	DINHEIRO
3	4	1	12 19.01.02	6875	RITA BRANCO	DINHEIRO
4	3	1	14 19.01.02	342	PAULO LOPES	MULTIBANCO
5	2	1	16 19.01.02	17485	DIOGO CORREIA	MULTIBANCO
6	1	1	14 19.01.02	17512	PEDRO SILVA	MULTIBANCO

Fig. 6 - View V_PAGAMENTOS

NRO_PROFESSOR	NOME	RUA	PORTA	CODIGO_POSTAL	LOCALIDADE	TELEFONE	EMAIL	NIF
1 P001	VASCO SILVA	RUA SANTIAGO DA BARRA	17	4900-357	RUA SANTIAGO DA BARRA	965789987	vascosilva@ipvc.pt	123456789
2 P002	ANA MIRANDA	CASTANHEIRA	54	4970-555	SÃO COSME E SÃO DAMIÃO	912343211	mirandaana@ipvc.pt	987654321
3 P003	ANTONIO ALVES	RUA DA LIBERDADE	7	4970-441	PRAÇA MUNICIPAL	933444555	a.alves@ipvc.pt	543219876

Fig. 7 - View V_PROFESSORES

NRO_SESSAO	DATA	HORA_INICIO	HORA_FIM	DISCIPLINA	PROFESSOR	SALA
1	4 19.11.01	11H00	13H00	ABD	ANTONIO ALVES	A1.1
2	3 18.12.02	10H00	12H00	ABD	ANTONIO ALVES	L1.2
3	2 19.01.02	14H00	16H00	IHM	ANA MIRANDA	S2.1
4	1 18.12.13	10H00	12H00	ASC	VASCO SILVA	L1.1

Fig. 8 - View V_SESSOES

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 ATUALIZAÇÃO 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(V 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 REGISTRADAS:**

```
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

- **LISTAR OS PROFESSORES QUE LECIONARAM EM DEZEMBRO DE 2018:**

```
CREATE VIEW V_PROFESSOR_SESSAO(NUM_SESSAO, DATA, DISCIPLINA, PROFESSOR)
AS
SELECT S.NUM_SESSAO, S.DATA, D.DESCRICAO, P.NOME
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;
```

```
SELECT DISTINCT PROFESSOR FROM V_PROFESSOR_SESSAO
WHERE DATA >= TO_DATE('01.12.2018','DD.MM.YYYY')
AND DATA < TO_DATE('01.01.2019','DD.MM.YYYY');
```

- **LISTAR AS DISCIPLINAS DAS SESSÕES A QUE CADA ALUNO ASSISTIU:**

```
SELECT DISTINCT A.NOME, V_D.NOME
FROM ALUNO A, V_DISCIPLINAS V_D, SESSAO S, ALUNO_SESSAO A_S
WHERE A.NUM_ALUNO = A_S.NUM_ALUNO AND A_S.NUM_SESSAO = S.NUM_SESSAO
AND S.ID_DISCIPLINA = V_D.ID_DISCIPLINA
ORDER BY A.NOME;
```

3. CONCLUSÃO

A concretização deste relatório e da criação da Base de Dados para a gestão de um centro de Explicações, foi uma experiência valiosa e gratificante, que constituiu uma fonte de significativa aprendizagem em múltiplas vertentes.

Além de nos ter permitido treinar e fazer uso de competências técnico-práticas na criação e desenvolvimento completamente autónomo de uma base de dados que se revelou intelectualmente estimulante, fez com que nos apercebesse-mos dos inesperados obstáculos e dificuldades que surgem quando se tenta desenvolver uma base de dados para suportar um determinado negocio, impelindo-nos a encontrar soluções que permitissem contornar as dificuldades e limitações com que nos deparamos.

No entanto, não obstante as limitações e algumas imperfeições fruto da inexperiência e competências ainda em maturação, julgo que logamos alcançar todos os objetivos pretendidos pela professora.

REFERÊNCIAS

[1] ...

[2] ...

[3] ...