

Relatório Final



Departamento de Engenharia Informática e de Sistemas

[Nome do curso]

Bases de Dados

2022/2023

Tema

Base de Dados do ISEC

Identificação dos Autores

Autores:

Nº Aluno	Prática Nome	Email
2021134104 P1	João Pedro Rodrigues Salgado	a2021134104@isec.pt
2021138502 P2	Guilherme de Sousa Camacho	a2021138502@isec.pt
2021141476 P2	Henrique Pires Marques	a2021141476@isec.pt
2021133564 P6	João Eduardo Oliveira Neves	a2021133564@isec.pt

Distribuição do esforço

Tarefa	João Pedro	Henrique Marques	Guilherme Camacho	João Neves
Estruturação, Analise e Organização Global	25%	25%	25%	25%
Planificação e Organização a nível Administrativo (Docentes, Alunos, Professores, Registros, Propinas, Previélegios, Gestão e Serviços)	30%	20%	30%	20%
Planificação e Organização a nível Escolar (Cursos, Departamentos, Disciplinas, Salas, Avaliações, Trabalhos)	20%	30%	20%	30%
Relações e Projeção da Base de Dados	30%	20%	30%	20%
Realização e Desenvolvimento do Projeto	40%	15%	25%	20%
Relatório e Planificação Final	15%	40%	20%	25%

Cap 1 - Introdução

O ISEC, tal como o nome indica, é o Instituto Superior de Educação de Engenharia de Coimbra. Por ser um instituto de grande abrangência tanto a nível técnico como profissional naturalmente necessita de uma base de dados complexa, independente e prática.

O objetivo deste projeto é estudar e planificar aquilo que será uma possível base de dados melhor optimizada para o ISEC tornando-a mais independente do IPC do que é hoje

No capítulo 2, faz-se um enquadramento da relevância da proposta no âmbito de implementar uma base de dados que optimize a gestão do ISEC, descreve-se a situação actual, os principais problemas encontrados e, por fim, descreve-se a proposta que irá permitir solucionar esses problemas e atingir o objectivo desejado.

No capítulo 3 apresenta-se a análise de dados necessária para a solução proposta, o modelo conceptual da base de dados, descrevendo-se todas as entidades e os relacionamentos existentes entre elas.

No capítulo 4, apresenta-se o modelo físico (ou diagrama de tabelas) da base de dados que vai servir de suporte ao sistema de informação proposto, assim como o respectivo script de criação da base de dados.

Por fim, no capítulo 5 são apresentadas conclusões do trabalho desenvolvido.

Cap 2 - Enquadramento da Proposta

Neste capítulo faz-se um enquadramento da proposta de organização da Base de Dados do ISEC. Na secção 2.1 descreve-se o funcionamento actual da base de dados, apresentando-se na secção 2.2 o conjunto de problemas existentes actualmente na mesma e que se pretendem solucionar. A secção 2.3 descreve a proposta de solução que permitirá solucionar os problemas existentes e satisfazer as necessidades encontradas.

Sec 2.1 - Diagnóstico da Situação Atual

Atualmente a base de dados não é apenas gerida pelo ISEC mas sim também pelo IPC - Instituto Politécnico de Coimbra. Tomamos por exemplo a questão dos emolumentos e propinas, todos estes são geridos pelo IPC em si, sendo que o ISEC tem pouco (ou nenhum) controlo sobre. Este tipo de gestão poderá trazer alguns problemas e inconsistências, que serão apresentados na secção seguinte.

Sec 2.2 - Problemas encontrados

Após a análise do modelo atual (descrito na secção acima), identificamos que todas as atuais dependências que o ISEC tem ao IPC originam algumas limitações e atraso em momentos que possivelmente com uma maior autonomia e melhor organização não existiriam. A possível incompatibilidade na forma que o IPC e ISEC escrevem os seus de dados poderá originar falsos resultados em consultas quando estes utilizam diferentes campos como chave primária (elementos identificadores que normalmente são utilizados para busca de informação).

Sec 2.3 - Descrição da Solução Proposta

A nossa solução passará por unificar a base de dados, incluindo os tópicos geridos pelo IPC na base de dados interna pelo ISEC, atribuindo-lhe assim todo o controlo sobre a mesma, facilitando o seu acesso. Esta solução ajudará também a que seja mais fácil os utilizadores do ISEC terem um maior contacto e maior "input" sobre os atributos das entidades, já que não existe uma relação de dependência, tanto a nível de disponibilidade de tempo como de recursos.

Cap 3 - Análise de Dados

Neste capítulo pretende-se apresentar o modelo conceptual da base de dados e descrever detalhadamente todas as entidades e relacionamentos envolvidos na solução proposta.

Sec 3.1 - Entidades

Nesta secção vão ser descritas todas as Entidades relevantes para a organização do ISEC. Após analisarmos o sistema de organização, concluímos a necessidade de utilizar as seguintes entidades:

- Curso
- Utilizador do ISEC
- Disciplina
- Funcionários
- Professor
- Departamento
- Sala
- Avaliação
- Trabalho
- Aluno
- Aula
- Morada
- Registo
- Serviço
- Cargo
- Privilégios
- Categoria
- Propinas

Sec 3.1.1 - Entidade Curso

A entidade Curso será a responsável por armazenar os dados do curso.
Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
Num_Curso	Numérico (8 dígitos)	Código interno único atribuído a cada utilizador. É um número de 8 dígitos gerado sequencialmente. Ex. 922233002
Nome	40 Caracteres	Nome do Curso. Ex. Licenciatura em Engenharia Informática
Num_ECTS	Numérico	Número de ECTS do curso
Num_Disciplinas	Numérico	Número de Disciplinas
Num_Anos	Numérico	Número de anos do curso

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Valores Nulos?	Valores Únicos?	Observações
Num_Curso	N	S	Identificador (chave primária), não admite nulos. Não existem dois utilizadores com o mesmo número de utilizador.
Nome	N	S	Não admite valor nulo.
Num_ECTS	N	N	Não admite valor nulo.
Num_Disciplinas	N	N	Não admite valor nulo.
Num_Anos	N	N	Não admite valor nulo.

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação
Possui	1:N	Disciplina	Obrigatória

Sec 3.1.2 - Entidade Utilizador do ISEC

A entidade Utilizador do Isec será a responsável por armazenar os dados comuns entre todos os utilizadores, sejam eles alunos, professores, funcionários não docentes, entre outros. Esta entidade representa qualquer pessoa que faça parte do ISEC. É inserido um novo registo nesta entidade sempre que é registado algum utilizador nas plataformas feitas para esse efeito.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
Num_Utilizador	Numérico (8 dígitos)	Código interno único atribuído a cada utilizador. É um número de 8 dígitos gerado sequencialmente. Ex. 92233002
Nome	40 Caracteres	Nome completo do utilizador. Ex. Jorge Francisco Costa
Data_Eentrada	Data	Data na qual foi inserido o utilizador. Ex. 10/09/2020 (dd/mm/aa)
Nacionalidade	20 Caracteres	Nacionalidade do utilizador Ex. Português

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
Num_Utilizador	N	S	Identificador (chave primária), não admite nulos. Não existem dois utilizadores com o mesmo número de utilizador.
Nome	N	N	Não admite valor nulo.
Data_Entrada	N	N	Não admite valor nulo.
Nacionalidade	N	N	Não admite valor nulo.

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação
Ativo			Obrigatória

Sec 3.1.3 - Entidade Disciplina

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
Num_Disciplina	Numérico (8 dígitos)	Código interno único atribuído a cada aluno. É um número de 8 dígitos gerado sequencialmente. Ex. 92281002

Nome	Texto	Nome da disciplina

Semestre	Numérico	Número destinado a indicar em qual dos dois semestres a disciplina pertence
Num_ECTS	Numérico	Número de ECTSs que uma disciplina atribui

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
Num_Disciplina	N	S	Identificador (chave primária), não admite nulos.
Nome	N	N	Não admite nulos.
Semestre	N	N	Não admite nulos.
Num_ECTS	N	N	Não admite nulos.

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação
Professor Responsável	N:1	Professor	Obrigatória
Enunciar	1:N	Trabalho	Professor
Avaliação	1:N	Avaliação	Disciplina

Sec 3.1.4 - Entidade Funcionários

A entidade Funcionários representa a informação relativa aos utilizadores que são funcionários do ISEC. Esta entidade não representa todos os utilizadores que existem, mas apenas aqueles que são funcionários. Todos os utilizadores que se registem como funcionários do ISEC estão registados nesta entidade (são registados no momento de criação do utilizador ou num momento futuro em que o utilizador se torne um funcionário). Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição		
Num Funcionário	Numérico (8 dígitos)	Código interno único atributo a cada administrador. É um número de 8 dígitos gerado sequencialmente. Ex. 92281002		

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Valores Nulos? Únicos?	Observações		
Num_administrador	N	S	Identificador (chave primária), não admite nulos.	

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação
Ativo			Obrigatória

Sec 3.1.5 - Entidade Professor

A entidade Professor representa a informação relativa aos utilizadores que são professores. Esta entidade não representa todos os utilizadores que existem. É registado um novo professor nesta entidade sempre que é registado um utilizador com funções de professor (ou que venha a escolher futuramente essa mesma função).
Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição		
Num_Professor	Numérico (8 dígitos)	Código interno único atributo a cada professor. É um número de 8 dígitos gerado sequencialmente. Ex. 92281002		

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Valores Nulos? Únicos?			Observações
Num_Professor	N	S	Identificador (chave primária), não admite nulos.	

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação
Posição	1 : N	Departamento	Obrigatória
Trabalha	1:N	Leciona	Professor
Dá	1:N	Aula	Aula

Sec 3.1.6 - Entidade Departamento

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
N_Departamento	Numérico	Código único atribuído a cada departamento.
Nome	40 Caracteres	Nome do departamento
N_Andares	Numérico	Número de andares no departamento

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
N_Departamento	N	S	Identificador (chave primária), não admite nulos. Não existem dois departamentos com o mesmo código.
Nome	N	N	Não admite nulos.
N_Andares	N	N	Não admite nulos.

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação
Localiza-se	1:N	Aula	Departamento

Sec 3.1.7 - Entidade Sala

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
Num Sala	Numérico (8 dígitos)	Código único atribuído a cada sala. Ex: 99654921
Nome	40 Caracteres	Nome dado a sala Ex: Anfiteatro Guerra Pratas

Capacidade	Numérico	Dia direcionado para a realização da mesma.	
Departamento	20 Caracteres	Departamento onde a sala se encontra	

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
Num Sala	N	S	Identificador (chave primária), não admite nulos. Não existem duas salas com o mesmo código.
Nome	N	N	Não admite nulos.
Capacidade	N	N	Não admite nulos.
Departamento	N	N	Não admite nulos.

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória

Sec 3.1.8 - Entidade Avaliação

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
Código Avaliação	Numérico	Código único da avaliação
Data	Data	Data da realização da avaliação
Cotação	Numérico	Cotação da avaliação (nota máxima) Ex: 20
Duração	Numérico	Duração da avaliação (em minutos)

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
Código Avaliação	N	S	Chave primária. Não admite nulos
Data	N	N	Não admite nulos.
Cotação	N	N	Não admite nulos.
Duração	N	N	Não admite nulos.

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
Marcação	1:N	Realização	Avaliação

Agendar	N:1	Disciplina	
---------	-----	------------	--

Sec 3.1.9 - Entidade Trabalho

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
Código Trabalho	Numérico (10 dígitos)	Código único atributo a cada trabalho. Ex: 9965492188
Data	Data & Tempo	Data estipulada para a entrega do trabalho
Cotação	Numérico	Cotação do trabalho (Nota máxima)

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos	Valores Únicos?	Observações
Código Trabalho	N	S	Identificador (chave primária), não admite nulos. Não existem dois trabalhos com o mesmo código.
Data	N	N	Não admite nulos.
Cotação	N	N	Não admite nulos.

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação
Enunciar	N:1	Disciplina	Disciplina

Sec 3.1.10 - Entidade Aluno

A entidade Aluno representa a informação relativa aos alunos do ISEC. Esta entidade não representa todos os utilizadores que existem, mas alguns deles. Todos os autores que escreveram livros que a livraria vende e/ou já vendeu, estão registados nesta entidades. É inserido um novo aluno nesta Entidade sempre que é registado um utilizador como aluno.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
Num_Aluno	Numérico (8 dígitos)	Código interno único atributo a cada aluno. É um número de 8 dígitos gerado sequencialmente. Ex. 92281002

ECTs	Numérico (3 dígitos)	Número de ECTs que o aluno possui no momento Ex. 152
Media	Numérico (2 dígitos e 1 casa decimal)	Média de curso do aluno Ex. 11,5
Ramo	20 Caractéres	Ramo escolhido pelo aluno

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
Num_Aluno	N	S	Identificador (chave primária), não admite nulos.
ECTs	N	N	Não admite nulos.
Media	N	N	Não admite nulos.
Ramo	S	N	Pode admitir nulos.

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
Gestão	1:N	Propinas	Propinas
Inscrito	1:N	Turma	Utilizador do ISEC
Ativo		Utilizador do ISEC	Utilizador do ISEC
Presença	1:N	Frequência	//
Realização	1:N	Trabalho	//

Sec 3.1.11 - Entidade Aula

A entidade Aula representa a informação relativa às aulas.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
Código Aula	Numérico	Código único da aula
Duração	Numérico	Duração da aula (em minutos)
Horário	Data e Hora	Horário da aula

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
Código Aula	N	S	Identificador (chave primária), não admite nulos.
Duração	N	N	Não admite nulos.
Horário	N	N	Não admite nulos.

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
Localiza-se	N:1	Departamento	Departamento
Dá	N:1	Professor	Aula
Frequenta	N:N	Aluno	Aula

Sec 3.1.12 - Entidade Morada

A entidade Morada representa a informação relativa à morada de um certo Utilizador do ISEC. É inserido um novo registo nesta Entidade sempre que é registado um novo utilizador, já que é obrigatório cada utilizador ter pelo menos uma morada associada.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
Código_Morada	Numérico (8 dígitos)	Código interno único atributo a cada morada. É um número de 8 dígitos gerado sequencialmente. Ex. 92281002
País	20 Caracteres	Nome do país
Cidade	20 Caracteres	Nome da cidade
Código_Postal	8 Caracteres	Código Postal da morada Ex. 3030-199, Coimbra
Rua	Texto	Nome Completo da Rua Ex. Rua Pedro Nunes

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
Código_Morada	N	N	Identificador (chave primária), não admite nulos.
País	N	N	Não admite valor nulo.

Cidade	N	N	Não admite valor nulo.
Código_Postal	N	N	Não admite valor nulo.
Rua	N	N	Não admite valor nulo.

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação
Habita	1 : N	Utilizador do ISEC	Utilizador do ISEC e Morada

Sec 3.1.13 - Entidade Registos

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
ID Registo	Numérico	Código único atribuíto a cada Registo.
Tarefa	20 Caracteres	Ação tomada.
Log	Data e Tempo	Dia e hora da ação.

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
------------------	---------------	-----------------	-------------

ID Registo	N	S	Identificador (chave primária), não admite nulos. Não existem duas turmas com o mesmo código.
Tarefa	N	N	Não admite nulos.
Log	N	N	Não admite nulos.

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação
Ação	N:N	Utilizador do ISEC	Utilizador do ISEC

Sec 3.1.14 - Entidade Cargo

A entidade Cargo representa a posição estática e hierárquica que a pessoa ocupa dentro da estrutura organizacional em que a mesma se encontra.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
Codigo_Cargo	Numérico (8 dígitos)	Código único atribuído a cada cargo. É um número de 8 dígitos gerado sequencialmente.
Nome	20 Caracteres	Designação da posição no departamento ou ambiente que a pessoa está envolvida. Ex. Administrativo

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
Codigo_Cargo	N	S	Identificador (chave primária), não admite nulos. Não existem dois cargos com o mesmo código.
Designação	N	N	Valor não nulo

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação
Pertence	N:N	Funcionários	Cargo e Funcionários
Contém	1:N	Privilegios	-

Sec 3.1.15 - Entidade Privilégios

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição

ID	Numérico (8 dígitos)	Código único atributo a cada privilégio
ID_Tabela	Numérico	Id da tabela onde estão presentes os privilégios
Editar	Booleano	Indica se tem/não tem permissão para editar
Ver	Booleano	Indica se tem/não tem permissão para ver

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
ID	N	S	Identificador (chave primária), não admite nulos. Não existem dois autores com o mesmo código.
ID_Tabela	N	N	Não admite nulos
Categoria	N	N	Não admite nulos

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação
tem	N:1	Serviço	Obrigatória
contém	N:1	Cargo	-

Sec 3.1.16 - Entidade Propinas

A entidade Propinas, como o nome indica são todas as características das propinas que cada aluno entra em contacto // referir depois a parte das posições depois de esclarecido

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
Código_Propina	Numérico (8 dígitos)	Código único atribuído a cada pagamento. É um número de 8 dígitos gerado sequencialmente.
Montante	Monetário	Quantidade de dinheiro necessário para pagar a propina
Data Limite	Data	Data com dia/mês/ano até ao ponto que a propina seja vencida
Vencimento	Booleana	Serve para definir o estado da propina, vencida ou não vencida, dependendo se a data atual ultrapassa ou não a Data Limite
Referência	Numérico	Numero do pagamento gerado no momento
Entidade	Numérico	Instituição para qual irá ser paga a propina
Estado Pagamento	Booleana	Estados a definir o estado do pagamento, pago ou não pago.

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Valores Nulos?	Valores Únicos?	Observações
Código_Propina	N	S	Identificador (chave primária), não admite nulos. Não existem duas propinas com o mesmo código.

Montante	N	N	N	Não admite nulos.
Data Limite	N	N	N	Não admite nulos.
Vencimento	N	N	N	Não admite nulos
Referência	N	S	S	Não admite nulos
Entidade	N	N	N	Não admite nulos
Estado Pagamento	N	N	N	Não admite nulos

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação
Gestão	N:1	Aluno	Obrigatória Aluno e Propinas

Sec 3.1.17 - Entidade Categoria

A entidade Categoria, como o nome indica são todas as características das categorias em que cada professor se insere

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
Código_Categoria	Numérico (8 dígitos)	Código único atributo a cada Categoria

Nome	40 Caracteres	Nome da Categoria
Função	Texto	Função desempenhada pela categoria

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
Código Categoria	N	S	Identificador (chave primária), não admite nulos. Não existem duas categorias com o mesmo código.
Nome	N	N	Não admite nulos.
Função	N	N	Não admite nulos.

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
Faz parte	1:N	Professor	Professor

Sec 3.2 - Relacionamentos

Nesta secção são descritos todos os relacionamentos existentes entre as várias entidades. Após uma análise aprofundada do negócio de venda de livros, constatou-se a necessidade dos seguintes relacionamentos:

- Habita

- Possui
- Ação
- Materia
- Faz parte
- Ensina
- Responsável
- Enuncia
- Atribuído
- Tem
- Contém
- Pertence
- Marcação
- Agenda
- Faz
- Presença
- Frequenta
- Dá
- Posição/Cargo
- Está
- Localiza-se
- Gestão

Sec 3.2.1 - Relacionamento: Habita

Este relacionamento pretende expressar o relacionamento existente entre as Entidades Morada e Utilizador do ISEC. A intenção é atribuir a cada utilizador uma morada.

Após uma análise do funcionamento do ISEC, definiu-se as seguintes condições:

- Tem que existir um utilizador para poder haver uma morada associada ao mesmo
- É obrigatório cada utilizador ter uma morada

Tomando estas condições em consideração, definiram-se as seguintes características

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
----------	-------------	---------------	-------------	----------

Morada	S	1:N	S	Utilizador do ISEC
Observações				
• Uma morada tem que ter pelo menos um utilizador do isec • Podem haver mais utilizadores do isec com a mesma morada				

Sec 3.2.2 - Relacionamento: Ação

Este relacionamento pretende expressar o relacionamento existente entre as Entidades Registo e Utilizador do ISEC. A intenção é atribuir a cada Ação um Registo.

Após uma análise do funcionamento do ISEC, definiu-se as seguintes condições:

- Tem que existir um utilizador para poder haver uma morada associada ao mesmo
- É obrigatório cada utilizador ter uma morada

Tomando estas condições em consideração, definiram-se as seguintes características

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Registo	N	N:1	S	Utilizador do ISEC
Observações				
• Um utilizador pode (ou não) ter registo				

Sec 3.2.3 - Relacionamento: Faz parte

Este relacionamento pretende expressar o relacionamento existente entre as Entidades Professor e Categoria. A intenção é atribuir a cada professor uma categoria.

Após uma análise do funcionamento do ISEC, definiu-se as seguintes condições:

- Um professor tem que fazer parte de uma categoria

Tomando estas condições em consideração, definiram-se as seguintes características

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Professor	N	N:1	S	Categoria
Observações				
<ul style="list-style-type: none"> • Um professor tem que fazer parte de uma categoria • É necessária uma categoria para adicionar um professor 				

Sec 3.2.4 - Relacionamento: Ensina

Este relacionamento pretende expressar o relacionamento existente entre as Entidades/Relacionamentos Professor e Leciona. A intenção é atribuir a cada professor um ano no qual leciona

Após uma análise do funcionamento do ISEC, definiu-se as seguintes condições:

- Um professor tem que ter um ano no qual leciona

Tomando estas condições em consideração, definiram-se as seguintes características

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Leciona	S	N:1	S	Professor
Observações				
<ul style="list-style-type: none"> • Um professor tem que lecionar num certo ano 				

Sec 3.2.5 - Relacionamento: Responsável

Este relacionamento pretende expressar o relacionamento existente entre as Entidades/Relacionamentos Professor e Leciona. A intenção é atribuir a cada professor um ano no qual leciona e associá-lo a uma disciplina

Após uma análise do funcionamento do ISEC, definiu-se as seguintes condições:

- Um professor tem que ser responsável

Tomando estas condições em consideração, definiram-se as seguintes características

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Disciplina	N	N:1	S	Professor
Observações				
• Um professor pode ser responsável por uma disciplina • Uma disciplina tem, obrigatoriamente, que ter um responsável pela mesma				

Sec 3.2.6 - Relacionamento: Leciona

Este relacionamento pretende expressar o relacionamento existente entre as Entidades/Relacionamentos Disciplina e Professor. A intenção é atribuir a cada professor um ano no qual leciona

Após uma análise do funcionamento do ISEC, definiu-se as seguintes condições:

- Um professor tem que ser responsável

Tomando estas condições em consideração, definiram-se as seguintes características

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Professor	S	N : 1	S	Disciplina
Observações				

- Um professor tem de ter (pelo menos) uma disciplina
- Uma disciplina precisa de (pelo menos) um professor

Atributos relevantes do Relacionamento:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
Ano_Letivo	N	N	Ano letivo no qual o professor leciona a disciplina

Sec 3.2.7 - Relacionamento: Posição/Cargo no Departamento

Este relacionamento pretende expressar o relacionamento existente entre as Entidades Professor e Departamento. A intenção é atribuir a cada professor um Cargo no departamento.

Após uma análise do funcionamento do ISEC, definiu-se as seguintes condições:

- Um professor tem que ser responsável

Tomando estas condições em consideração, definiram-se as seguintes características

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Professor	S	N:1	N	Departamento

Observações

- Tem de haver um professor com cargo no departamento

Sec 3.2.8 - Relacionamento: Agenda

Este relacionamento pretende expressar o relacionamento existente entre as Entidades Disciplina e Avaliação Escrita. A intenção é atribuir uma avaliação escrita a cada disciplina.

Após uma análise do funcionamento do ISEC, definiu-se as seguintes condições:

- Tem de haver uma disciplina para haver uma avaliação escrita

Tomando estas condições em consideração, definiram-se as seguintes características

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Avaliação Escrita	N	N:1	S	Disciplina

Observações

- Tem de haver uma disciplina para poder haver uma avaliação escrita
- Não é necessário haver uma avaliação escrita numa disciplina

Sec 3.2.9 - Relacionamento: Enuncia

Este relacionamento pretende expressar o relacionamento existente entre as Entidades Disciplina e Trabalho. A intenção é atribuir um trabalho a uma disciplina.

Após uma análise do funcionamento do ISEC, definiu-se as seguintes condições:

- Tem de haver uma disciplina para haver um trabalho

Tomando estas condições em consideração, definiram-se as seguintes características

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Trabalho	N	N:1	S	Disciplina

Observações

- | |
|--|
| <ul style="list-style-type: none"> • Tem de haver uma disciplina para poder haver um trabalho • Não é necessário haver um trabalho numa disciplina |
|--|

Sec 3.2.10 - Relacionamento: Realização

Este relacionamento pretende expressar o relacionamento existente entre as Entidades Avaliação Escrita e Aluno, este é um relacionamento com atributos. A intenção é interligar as avaliações no ISEC e quem as realiza .

Após uma análise do funcionamento do ISEC, definiu-se as seguintes condições:

- A Realização está dependente da presença do aluno e da marcação da avaliação escrita.

Tomando estas condições em consideração, definiram-se as seguintes características

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Avaliação Escrita	S	1:1	S	Aluno
Observações				
<ul style="list-style-type: none"> • Uma realização de uma avaliação escrita somente gera um resultado • Um resultado pertence a um aluno. 				

Atributos relevantes do Relacionamento:

Nome do atributo	Aceita Valores Nulos? Únicos?	Observações
Data_Entrega	N	Não admite nulos
Realização	N	Não admite nulos

Sec 3.2.11 - Relacionamento: Gestão

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Propinas	S	N : 1	S	Aluno
Observações				

• Um aluno tem que ter propinas
• As propinas pertencem a um aluno (é necessitam da existência de um aluno)

Este relacionamento pretende expressar o relacionamento existente entre as Entidades Aluno e Propinas, no que concerne a quem gera o pagamento das propinas. A intenção é interligar o pagamento das propinas com a ação dos alunos.

Tomando estas condições em consideração, definiram-se as seguintes características

Sec 3.2.12 - Relacionamento: Realização

Este relacionamento pretende expressar o relacionamento existente entre as Entidades Aluno e Realização, no que concerne a quem está presente para a realização da Avaliação escrita. A intenção é interligar a realização de avaliações no ISEC e quem está presente.

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Realização	S	N:1	S	Aluno
Observações				

• A realização da prova é dependente da presença do aluno
• Um aluno tem que estar presente para realizar a prova.

Sec 3.2.13 - Relacionamento: Marcação

Este relacionamento pretende expressar o relacionamento existente entre as Entidades Avaliação Escrita e Realização, no que concerne a quem marca a Avaliação escrita. A intenção é interligar a realização de avaliações no ISEC com quem a marca.

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Realização	S	N:1	S	Avaliação Escrita
Observações				

- A realização da prova é dependente da marcação da Avaliação.
- Um aluno tem que se inscrever ou marcar para poder realizar a prova escrita.

Sec 3.2.14 - Relacionamento: Frequentia

Este relacionamento pretende expressar o relacionamento existente entre as Entidades Aluno e Aula, no que concerne a quem frequenta a aula. A intenção é interligar as aulas dadas no ISEC e quem vai assistir às mesmas.

Após uma análise do funcionamento do ISEC, definiu-se as seguintes condições:

- Um Aula é obrigada a ter aluno(s).
- Não é obrigatório o aluno ter uma aula.

Tomando estas condições em consideração, definiram-se as seguintes características

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Aula	N	N : N	S	Aluno
Observações				

- O Numero de Aluno vai ser sempre diferente
- A media os ECTs o ramo e o nome podem ser iguais

Sec 3.2.15 - Relacionamento: Da

Este relacionamento pretende expressar o relacionamento existente entre as Entidades Professor e Aula, no que concerne a quem leciona da aula. A intenção é interligar as aulas dadas no ISEC e quem está responsável pelas mesmas.

Após uma análise do funcionamento do ISEC, definiu-se as seguintes condições:

- Um Aula é obrigada a ter professor.
- Não é obrigatório o professor dar aula

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Aula	N	N : 1	S	Professor
Observações				

• Um professor tem apenas um numero de professor.
• Podem haver professores com o mesmo nome.

Sec 3.2.16 - Relacionamento: Está

Este relacionamento pretende expressar o relacionamento existente entre as Entidades Sala e Departamento, no que concerne ao local da aula. A intenção é interligar as aulas dadas no ISEC e a sua localização nos departamentos a estas destinados.

Após uma análise do funcionamento do ISEC, definiu-se as seguintes condições:

- Uma aula só pode ser dada se estiver dentro de um departamento
- Uma Departamento é obrigado a ter aulas
- É obrigatório uma aula ter departamento.

Tomando estas condições em consideração, definiram-se as seguintes características

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade

Aula	S	1 : N	S	Sala
Observações				
• O Num Departamento nunca pode ser o mesmo. • Um Departamento poderá ter varias salas.				

Sec 3.2.17 - Relacionamento: Localiza-se

Este relacionamento pretende expressar o relacionamento existente entre as Entidades Sala e Aula, no que concerne ao local da aula. A intenção é interligar as aulas dadas no ISEC e a sua localização nas salas a estas destinados.

Após uma análise do funcionamento do ISEC, definiu-se as seguintes condições:

- Não existe a possibilidade de uma aula se encontrar fora de uma Sala
- Uma sala pode ou não ter uma aula.
- Uma aula é dependente da sala.

Tomando estas condições em consideração, definiram-se as seguintes características

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Aula	N	1 : N	S	Sala
Observações				
• O código da aula nunca pode ser o mesmo. • Uma sala poderá ter varias aulas mas não ao mesmo tempo				

Sec 3.2.18 - Relacionamento: Possui

Este relacionamento pretende expressar o relacionamento existente entre as Entidades Curso e Disciplina. O objetivo é expressar que dentro de um curso pode haver varias disciplinas e que disciplinas distintas podem pertencer a um curso.

Após uma análise do funcionamento do ISEC, definiu-se as seguintes condições:

- Não existe a possibilidade de existirem disciplinas solitárias sem pertencerem a um curso.
- Um curso não pode existir sem ter disciplinas associadas.
- Não é requerido que existência prévia no curso, de uma disciplina, antes da sua inserção na base de dados.

Tomando estas condições em consideração, definiram-se as seguintes características

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Curso	S	1 : N	S	Disciplina
Observações:				
• O código da disciplina nunca pode ser o mesmo.				
• O atributo semestre vai ter sempre um valor, 1 ou 2				
• As disciplinas podem ter diversos ECTs.				

Sec 3.2.19 - Relacionamento: Atribuição

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Funcionários	S	N : N	S	Serviço
Observações:				
• Um funcionários tem que ter um (ou mais) serviço				
• Um serviço tem que ter uma (ou mais) funcionários				

Sec 3.2.20 - Relacionamento: Pertence

Tomando estas condições em consideração, definiram-se as seguintes características

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Funcionários	S	N : N	S	Cargo

Observações			
<ul style="list-style-type: none"> • Um funcionários tem que ter um (ou mais) cargo • Um cargo tem que ter uma (ou mais) funcionários 			

Sec 3.2.21 - Relacionamento: tem

Tomando estas condições em consideração, definiram-se as seguintes características

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Serviço	N	1 : N	N	Privilégios

Observações

- Um serviço pode não ter (ainda) privilégios
- Um privilégio pode não ter (ainda) serviço

Sec 3.2.22 - Relacionamento: contém

Tomando estas condições em consideração, definiram-se as seguintes características

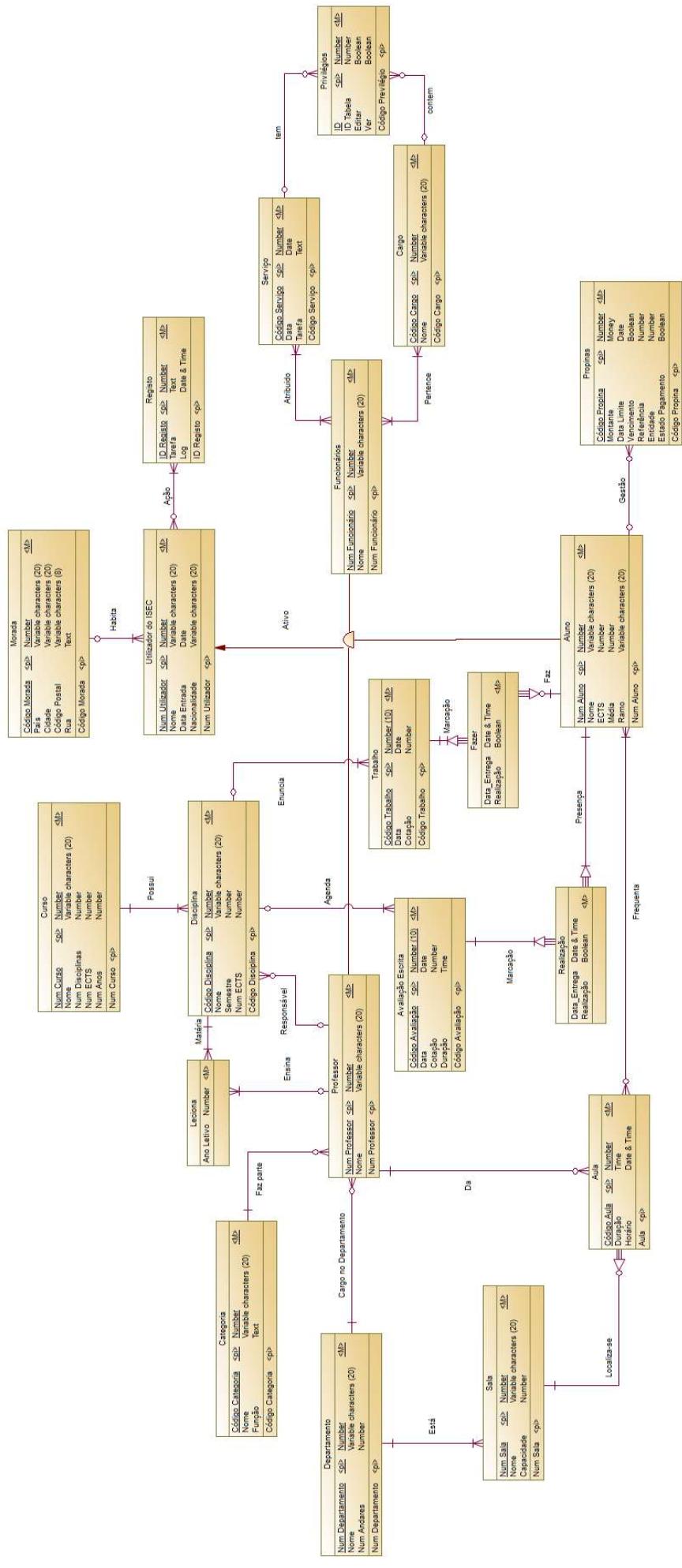
Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Cargo	N	1 : N	N	Privilégios

Observações

- Um cargo pode não ter (ainda) privilégios
- Um privilégio pode não ter (ainda) cargo

Sec 3.3 - Diagrama do Modelo Conceptual

O modelo conceptual de Entidade/Relacionamento completo é o seguinte:



Cap 4 - Modelo Físico

Neste capítulo documenta-se o modelo físico da base de dados e o script da criação da base de dados.

Sec 4.1 - Diagrama do Modelo Físico

O modelo Físico (ou diagrama de tabelas) completo com todas as tabelas é o seguinte:



Modelo Físico do Trabalho

Sec 4.2 - Script de criação da Base de Dados

```
/*=====
 * DBMS name: ORACLE Version 11g
 * Created on: 12/12/2022 05:07:37
 *=====*/
/*=====
 * ALTER TABLE ACAO
 * DROP CONSTRAINT FK_ACAO_ACAO2_UTILIZAD;
 *
 * ALTER TABLE ALUNO
 * DROP CONSTRAINT FK_ALUNO_INHERITAN_UTILIZAD;
 *
 * ALTER TABLE ATRIBUIDO
 * DROP CONSTRAINT FK_ATRIBUIDO_ATRIBUIDO_FUNCIONA;
 *
 * ALTER TABLE ATRIBUIDO
 * DROP CONSTRAINT FK_ATRIBUIDO_SERVICO;
 */
```

```
alter table AULA
drop constraint FK_AULA_DA_PROFESSO;

alter table AULA
drop constraint FK_AULA_LOCALIZA_SALA;

alter table AVALIACAO_ESCRITA
drop constraint FK_AVALIACA_AGENDA_DISCIPLI;

alter table DISCIPLINA
drop constraint FK_DISCIPLI_POSSUI_CURSO;

alter table DISCIPLINA
drop constraint FK_DISCIPLI_RESPONSAV_PROFESSO;

alter table FAZER
drop constraint FK_FAZER_FAZ_ALUNO;

alter table FAZER
```

```
drop constraint FK_FAZER_MARCACAO_TRABALHO;
```

```
alter table FREQUENTA
```

```
drop constraint FK_FREQUENT_FREQUENTA_AULA;
```

```
alter table FREQUENTA
```

```
drop constraint FK_FREQUENT_FREQUENTA_ALUNO;
```

```
alter table FUNCIONARIOS
```

```
drop constraint FK_FUNCIONA_INHERITAN_UTILIZAD;
```

```
alter table LECIONA
```

```
drop constraint FK_LECIONA_ENSINA_PROFESSO;
```

```
alter table LECIONA
```

```
drop constraint FK_LECIONA_MATERIA_DISCIPLI;
```

```
alter table PERTENCE
```

```
drop constraint FK_PERTENCE_PERTENCE_FUNCIONA;
```

```
alter table PERTENCE
drop constraint FK_PERTENCE_PERTENCE2_CARGO;

alter table PRIVILEGIOS
drop constraint FK_PRIVILEG_CONTEM_CARGO;

alter table PRIVILEGIOS
drop constraint FK_PRIVILEG_TEM_SERVICO;

alter table PROFESSOR
drop constraint FK_PROFESSO_CARGO_NO_DEPARTAM;

alter table PROFESSOR
drop constraint FK_PROFESSO_FAZ_PARTE_CATEGORI;

alter table PROFESSOR
drop constraint FK_PROFESSO_INHERITAN_UTILIZAD;

alter table PROPINAS
drop constraint FK_PROPINAS_GESTAO_ALUNO;
```

```
alter table REALIZACAO
drop constraint FK_REALIZAC_MARCACAO2_AVALIACA;

alter table REALIZACAO
drop constraint FK_REALIZAC_PRESENCA_ALUNO;

alter table SALA
drop constraint FK_SALA_ESTA_DEPARTAM;

alter table TRABALHO
drop constraint FK_TRABALHO_ENUNCIA_DISCIPLI;

alter table UTILIZADOR_DO_ISEC
drop constraint FK_UTILIZAD_HABITA_MORADA;

drop index ACAO2_FK;

drop index ACAO_FK;
```

```
drop table ACAO cascade constraints;
```

```
drop index INHERITANCE_1_FK;
```

```
drop table ALUNO cascade constraints;
```

```
drop index ATRIBUIDO2_FK;
```

```
drop index ATRIBUIDO_FK;
```

```
drop table ATRIBUIDO cascade constraints;
```

```
drop index LOCALIZA_SE_FK;
```

```
drop index DA_FK;
```

```
drop table AULA cascade constraints;
```

```
drop index AGENDA_FK;
```

```
drop table AVALIACAO_ESCRITA cascade constraints;

drop table CARGO cascade constraints;

drop table CATEGORIA cascade constraints;

drop table CURSO cascade constraints;

drop table DEPARTAMENTO cascade constraints;

drop index RESPONSAVEL_FK;

drop index POSSUI_FK;

drop table DISCIPLINA cascade constraints;

drop index FAZ_FK;

drop index MARCACAO_FK;
```

```
drop table FAZER cascade constraints;
```

```
drop index FREQUENTA2_FK;
```

```
drop index FREQUENTA_FK;
```

```
drop table FREQUENTA cascade constraints;
```

```
drop index INHERITANCE_2_FK;
```

```
drop table FUNCIONARIOS cascade constraints;
```

```
drop index ENSINA_FK;
```

```
drop index MATERIA_FK;
```

```
drop table LECIONA cascade constraints;
```

```
drop table MORADA cascade constraints;
```

```
drop index PERTENCE2_FK;
```

```
drop index PERTENCE_FK;
```

```
drop table PERTENCE cascade constraints;
```

```
drop index CONTEM_FK;
```

```
drop index TEM_FK;
```

```
drop table PRIVILEGIOS cascade constraints;
```

```
drop index INHERITANCE_3_FK;
```

```
drop index FAZ_PARTE_FK;
```

```
drop index POSICAO_CARGO_NO_DEPARTAMENTO_;
```

```
drop table PROFESSOR cascade constraints;
```

```
drop index GESTAO_FK;
```

```
drop table PROPINAS cascade constraints;
```

```
drop index PRESENCA_FK;
```

```
drop index MARCACAO2_FK;
```

```
drop table REALIZACAO cascade constraints;
```

```
drop table REGISTO cascade constraints;
```

```
drop index ESTA_FK;
```

```
drop table SALA cascade constraints;
```

```
drop table SERVICO cascade constraints;
```

```
drop index ENUNCIA_FK;
```

```
drop table TRABALHO cascade constraints;
```

```
drop index HABITA_FK;
```

```
drop table UTILIZADOR_DO_ISEC cascade constraints;
```

```
/*=====
 * Table: ACAO
 */
```

```
=====*/
create table ACAO
```

```
(  
    NUM_UTILIZADOR NUMBER      not null,  
    ID_REGISTO     NUMBER      not null,  
    constraint PK_ACAO primary key (NUM_UTILIZADOR, ID_REGISTO)  
);
```

```
/*=====
 * Index: ACAO_FK
 */
```

```
=====*/
create index ACAO_FK on ACAO (
```

```
NUM_UTILIZADOR ASC
```

```
);
```

```
/*=====
*/
```

```
/* Index: ACAO2_FK
```

```
/*=====
*/
```

```
create index ACAO2_FK on ACAO (
```

```
ID_REGISTO ASC
```

```
);
```

```
/*=====
*/
```

```
/* Table: ALUNO
```

```
*/
```

```
/*=====
*/
```

```
create table ALUNO
```

```
(  
    NUM_UTILIZADOR NUMBER      not null,  
    NUM_ALUNO      NUMBER      not null,  
    CODIGO_MORADA NUMBER,  
    UTI_NOME       VARCHAR2(20),  
    DATA_ENTRADA   DATE,
```

```
NACIONALIDADE  VARCHAR2(20),
NOME          VARCHAR2(20),
ECTS          NUMBER,
MEDIA          NUMBER,
RAMO          VARCHAR2(20),
constraint PK_ALUNO primary key (NUM_UTILIZADOR, NUM_ALUNO)
);

/*=====
 * Index: INHERITANCE_1_FK
 */
create index INHERITANCE_1_FK on ALUNO (
    NUM_UTILIZADOR ASC
);

/*=====
 * Table: ATRIBUIDO
 */
create table ATRIBUIDO
(
```

```

NUM_UTILIZADOR NUMBER          not null,
NUM_FUNCIONARIO  NUMBER          not null,
CODIGO_SERVICO   NUMBER          not null,
constraint PK_ATRIBUIDO primary key (NUM_UTILIZADOR, NUM_FUNCIONARIO, CODIGO_SERVICIO)
);

/*=====
 * Index: ATRIBUIDO_FK
 *=====
 create index ATRIBUIDO_FK on ATRIBUIDO (
    NUM_UTILIZADOR ASC,
    NUM_FUNCIONARIO ASC
);

/*=====
 * Index: ATRIBUIDO2_FK
 *=====
 create index ATRIBUIDO2_FK on ATRIBUIDO (
    CODIGO_SERVICO ASC
);

```

```
/*=====
 * Table: AULA
 */
=====*/
create table AULA
(
    NUM_SALA      NUMBER      not null,
    CODIGO_AULA   NUMBER      not null,
    NUM_UTILIZADOR NUMBER      not null,
    NUM_PROFESSOR NUMBER      not null,
    DURACAO       DATE,
    HORARIO       DATE,
    constraint PK_AULA primary key (NUM_SALA, CODIGO_AULA)
);

/*=====
 * Index: DA_FK
 */
=====*/
create index DA_FK on AULA(
    NUM_UTILIZADOR ASC,
```

```
NUM_PROFESSOR ASC
```

```
);
```

```
/*=====
=====*/

```

```
/* Index: LOCALIZA_SE_FK
   */

```

```
/*=====
=====*/

```

```
create index LOCALIZA_SE_FK on AULA(

```

```
NUM_SALA ASC
```

```
);

```

```
/*=====
=====*/

```

```
/* Table: AVALIACAO_ESCRITA
   */

```

```
/*=====
=====*/

```

```
create table AVALIACAO_ESCRITA
```

```
(
```

```
CODIGO_TRABALHO2 NUMBER(10)      not null,
```

```
CODIGO_DISCIPLINA NUMBER,
```

```
DATA DATE,
```

```
COTACAO NUMBER,
```

```
DURACAO DATE,
```

```

constraint PK_AVALIACAO_ESCRITA primary key (CODIGO_TRABALHO2)
);

/*=====
 * Index: AGENDA_FK
 */

create index AGENDA_FK on AVALIACAO_ESCRITA(
    CODIGO_DISCIPLINAASC
);

/*=====
 * Table: CARGO
 */

create table CARGO
(
    CODIGO_CARGO      NUMBER          not null,
    NOME              VARCHAR2(20),
);

constraint PK_CARGO primary key (CODIGO_CARGO)
);

```

```

=====
/* Table: CATEGORIA */
=====

create table CATEGORIA
(
    (
        CODIGO_CATEGORIA   NUMBER      not null,
        NOME                VARCHAR2(20),
        FUNCAO              CLOB,
        constraint PK_CATEGORIA primary key (CODIGO_CATEGORIA)
    );
);

=====
/* Table: CURSO */
=====

create table CURSO
(
    (
        NUM_CURSO          NUMBER      not null,
        NOME                VARCHAR2(20),
        NUM_DISCIPLINAS    NUMBER,
        NUM_ECTS            NUMBER,
        ...
    );
);

```

```

NUM_ANOS      NUMBER,
constraint PK_CURSO primary key (NUM_CURSO)
);

/*=====
 * Table: DEPARTAMENTO
 *=====
create table DEPARTAMENTO
(
    NUM_DEPARTAMENTO  NUMBER
        not null,
    NOME              VARCHAR2(20),
    NUM_ANDARES      NUMBER,
constraint PK_DEPARTAMENTO primary key (NUM_DEPARTAMENTO)
);

/*=====
 * Table: DISCIPLINA
 *=====
create table DISCIPLINA
(

```

```

CÓDIGO_DISCIPLINA NUMBER          not null,
NUM UTILIZADOR   NUMBER,
NUM PROFESSOR    NUMBER,
NUM CURSO        NUMBER          not null,
NOME             VARCHAR2(20),
SEMESTRE         NUMBER,
NUM ECTS         NUMBER,
constraint PK_DISCIPLINA primary key (CÓDIGO_DISCIPLINA)

);

/*=====
 * Index: POSSUI_FK
 *=====
 create index POSSUI_FK on DISCIPLINA(
    NUM_CURSO ASC
);

/*=====
 * Index: RESPONSAVEL_FK
 *=====

```

```

create index RESPONSAVEL_FK on DISCIPLINA (
    NUM_UTILIZADOR ASC,
    NUM_PROFESSOR ASC
);

/*=====
 * Table: FAZER
 *=====
 create table FAZER
(
    CODIGO_TRABALHO NUMBER(10)      not null,
    NUM_UTILIZADOR NUMBER           not null,
    NUM_ALUNO        NUMBER           not null,
    DATA_ENTREGA     DATE,
    REALIZACAO       SMALLINT        not null,
    constraint PK_FAZER primary key (CODIGO_TRABALHO, NUM_UTILIZADOR, NUM_ALUNO)
);

/*=====
 * Index: MARCACAO_FK
 *=====

```

```

/*
create index MARCACAO_FK on FAZER (
    CODIGO_TRABALHO ASC
);

/*
/* Index: FAZ_FK
*/
create index FAZ_FK on FAZER (
    NUM_UTILIZADOR ASC,
    NUM_ALUNO ASC
);

/*
/* Table: FREQUENTA
*/
create table FREQUENTA
(
    NUM_SALA      NUMBER      not null,
    CODIGO_AULA   NUMBER      not null,

```

```

NUM_UTILIZADOR NUMBER      not null,
NUM_ALUNO        NUMBER      not null,
constraint PK_FREQUENTA primary key (NUM_SALA, CODIGO_AULA, NUM_UTILIZADOR, NUM_ALUNO)
);

/*=====
 * Index: FREQUENTA_FK
 */

create index FREQUENTA_FK on FREQUENTA(
    NUM_SALA ASC,
    CODIGO_AULA ASC
);

/*=====
 * Index: FREQUENTA2_FK
 */

create index FREQUENTA2_FK on FREQUENTA(
    NUM_UTILIZADOR ASC,
    NUM_ALUNO ASC
);

```

```

/*
 * Table: FUNCIONARIOS
 */
create table FUNCIONARIOS
(
    NUM_UTILIZADOR NUMBER          not null,
    NUM_FUNCIONARIO NUMBER          not null,
    CODIGO_MORADA  NUMBER,
    UTI_NOME        VARCHAR2(20),
    DATA_ENTRADA   DATE,
    NACIONALIDADE VARCHAR2(20),
    NOME           VARCHAR2(20),
constraint PK_FUNCIONARIOS primary key (NUM_UTILIZADOR, NUM_FUNCIONARIO)
);

/*
 * Index: INHERITANCE_2_FK
 */
create index INHERITANCE_2_FK on FUNCIONARIOS (

```

```
NUM_UTILIZADOR ASC
```

```
);
```

```
/*=====
 */
```

```
/* Table: LECIONA
```

```
/*===== */
```

```
create table LECIONA
```

```
(  
    CODIGO_DISCIPLINA NUMBER      not null,  
    NUM_UTILIZADOR   NUMBER,  
    NUM_PROFESSOR    NUMBER,  
    ANO_LETIVO       NUMBER      not null  
)
```

```
/*===== */
```

```
/* Index: MATERIA_FK
```

```
/*===== */
```

```
create index MATERIA_FK on LECIONA (  
    CODIGO_DISCIPLINA ASC  
)
```

```

/*
 * Index: ENSINA_FK
 */
create index ENSINA_FK on LECIONA (
    NUM_UTILIZADOR ASC,
    NUM_PROFESSOR ASC
);

/*
 * Table: MORADA
 */
create table MORADA
(
    CODIGO_MORADA NUMBER not null,
    PAIS VARCHAR2(20),
    CIDADE VARCHAR2(20),
    CODIGO_POSTAL VARCHAR2(8),
    RUA CLOB,
    constraint PK_MORADA primary key (CODIGO_MORADA)
)

```

```

);

/*=====
 * Table: PERTENCE
 *=====
create table PERTENCE

(
    NUM_UTILIZADOR NUMBER      not null,
    NUM_FUNCIONARIO NUMBER      not null,
    CODIGO_CARGO   NUMBER      not null,
constraint PK_PERTENCE primary key (NUM_UTILIZADOR, NUM_FUNCIONARIO, CODIGO_CARGO)
);

/*=====
 * Index: PERTENCE_FK
 *=====
create index PERTENCE_FK on PERTENCE (
    NUM_UTILIZADOR ASC,
    NUM_FUNCIONARIO ASC
);

```

```

/*
 * Index: PERTENCE2_FK
 */
create index PERTENCE2_FK on PERTENCE (
    CODIGO_CARGO ASC
);

/*
 * Table: PRIVILEGIOS
 */
create table PRIVILEGIOS

(
    ID          NUMBER      not null,
    CODIGO_CARGO  NUMBER,
    CODIGO_SERVICO  NUMBER,
    ID_TABELA    NUMBER,
    EDITAR      SMALLINT,
    VER         SMALLINT,
constraint PK_PRIVILEGIOS primary key (ID)

```

```
);

/*=====
 * Index: TEM_FK
 */

/*=====
 * create index TEM_FK on PRIVILEGIOS (
 * CODIGO_SERVICIO ASC
 *);

/*=====
 * Index: CONTEM_FK
 */

/*=====
 * create index CONTEM_FK on PRIVILEGIOS (
 * CODIGO_CARGO ASC
 *);

/*=====
 * Table: PROFESSOR
 */

/*=====
 * create table PROFESSOR
```

```

(
    NUM_UTILIZADOR      NUMBER          not null,
    NUM_PROFESSOR       NUMBER          not null,
    NUM_DEPARTAMENTO   NUMBER          not null,
    CODIGO_CATEGORIA   NUMBER          not null,
    CODIGO_MORADA      NUMBER,
    UTI_NOME            VARCHAR2(20),
    DATA_ENTRADA        DATE,
    NACIONALIDADE      VARCHAR2(20),
    NOME                VARCHAR2(20),
constraint PK_PROFESSOR primary key (NUM_UTILIZADOR, NUM_PROFESSOR)
);

/*
* Index: POSICAO_CARGO_NO_DEPARTAMENTO_
*/
create index POSICAO_CARGO_NO_DEPARTAMENTO_on PROFESSOR (
    NUM_DEPARTAMENTO ASC
);

```

```

/*
 * Index: FAZ_PARTE_FK
 */
create index FAZ_PARTE_FK on PROFESSOR (
    CODIGO_CATEGORIA_ASC
);

/*
 * Index: INHERITANCE_3_FK
 */
create index INHERITANCE_3_FK on PROFESSOR (
    NUM_UTILIZADOR_ASC
);

/*
 * Table: PROPINAS
 */
create table PROPINAS
(
    CODIGO_PROPINA NUMBER          not null,

```

```
NUM_UTILIZADOR NUMBER,
NUM_ALUNO NUMBER,
MONTANTE NUMBER(8,2),
DATA_LIMITE DATE,
VENCIMENTO SMALLINT,
REFERENCIA NUMBER,
ENTIDADE NUMBER,
ESTADO_PAGAMENTO SMALLINT,
constraint PK_PROPINAS primary key (CODIGO_PROPINA)
);

/*=====
 * Index: GESTAO_FK
 *=====
 create index GESTAO_FK on PROPINAS (
    NUM_UTILIZADOR ASC,
    NUM_ALUNO ASC
);
/*=====*/
```

```
/* Table: REALIZACAO
```

```
/*=====*/
```

```
create table REALIZACAO
```

```
(  
    CODIGO_TRABALHO2 NUMBER(10)      not null,  
    NUM_UTILIZADOR   NUMBER          not null,  
    NUM_ALUNO        NUMBER          not null,  
    DATA_ENTREGA     DATE,  
    REALIZACAO       SMALLINT       not null,  
    constraint PK_REALIZACAO primary key (CODIGO_TRABALHO2, NUM_UTILIZADOR, NUM_ALUNO)  
);
```

```
/*=====*/
```

```
/* Index: MARCACAO2_FK
```

```
/*=====*/  
create index MARCACAO2_FK on REALIZACAO (  
    CODIGO_TRABALHO2 ASC  
);
```

```
/*=====*/
```

```
/* Index: PRESENCA_FK */  
-----*/  
  
create index PRESENCA_FK on REALIZACAO (  
    NUM_UTILIZADOR ASC,  
    NUM_ALUNO ASC  
);  
  
-----*/  
  
/* Table: REGISTO */  
-----*/  
  
create table REGISTO  
(  
    ID_REGISTO      NUMBER      not null,  
    TAREFA          CLOB,  
    LOG             DATE,  
    constraint PK_REGISTO primary key (ID_REGISTO)  
);  
  
-----*/  
  
/* Table: SALA */  
-----*/
```

```
=====*/
create table SALA
(
    NUM_SALA      NUMBER      not null,
    NUM_DEPARTAMENTO  NUMBER      not null,
    NOME          VARCHAR2(20),
    CAPACIDADE    NUMBER,
    constraint PK_SALA primary key (NUM_SALA)
);

=====*/
/* Index: ESTA_FK */
=====*/
create index ESTA_FK on SALA(
    NUM_DEPARTAMENTO ASC
);

=====*/
/* Table: SERVICO */
=====*/
```

```
create table SERVICO
(
    (
        CODIGO_SERVICO      NUMBER      not null,
        DATA                DATE,
        TAREFA              CLOB,
        constraint PK_SERVICO primary key (CODIGO_SERVICO)
    );
);

/*=====
 * Table: TRABALHO
 *=====
 create table TRABALHO

(
    (
        CODIGO_TRABALHO    NUMBER(10)   not null,
        CODIGO_DISCIPLINA  NUMBER,
        DATA                DATE,
        COTACAO             NUMBER,
        constraint PK_TRABALHO primary key (CODIGO_TRABALHO)
    );
);
```

```

/*
 * Index: ENUNCIA_FK
 */
create index ENUNCIA_FK on TRABALHO (
    CODIGO_DISCIPLINA ASC
);

/*
 * Table: UTILIZADOR_DO_ISEC
 */
create table UTILIZADOR_DO_ISEC

(
    NUM_UTILIZADOR NUMBER          not null,
    CODIGO_MORADA  NUMBER,
    NOME           VARCHAR2(20),
    DATA_ENTRADA   DATE,
    NACIONALIDADE VARCHAR2(20),
    constraint PK_UTILIZADOR_DO_ISEC primary key (NUM_UTILIZADOR)
);

```

```
=====
/* Index: HABITA_FK
 */
=====*/
create index HABITA_FK on UTILIZADOR_DO_ISEC (
    CODIGO_MORADA ASC
);

alter table ACAO
add constraint FK_ACAO_ACAO_UTILIZAD foreign key (NUM_UTILIZADOR)
references UTILIZADOR_DO_ISEC (NUM_UTILIZADOR);

alter table ACAO
add constraint FK_ACAO_ACAO2_REGISTO foreign key (ID_REGISTO)
references REGISTO (ID_REGISTO);

alter table ALUNO
add constraint FK_ALUNO_INHERITAN_UTILIZAD foreign key (NUM_UTILIZADOR)
references UTILIZADOR_DO_ISEC (NUM_UTILIZADOR);

alter table ATRIBUIDO
```

```
add constraint FK_ATRIBUID_ATRIBUIDO_FUNCIONA foreign key (NUM_UTILIZADOR, NUM_FUNCIONARIO)
references FUNCIONARIOS (NUM_UTILIZADOR, NUM_FUNCIONARIO);

alter table ATRIBUIDO
add constraint FK_ATRIBUID_ATRIBUIDO_SERVICO foreign key (CODIGO_SERVICO)
references SERVICO (CODIGO_SERVICO);

alter table AULA
add constraint FK_AULA_DA_PROFESSO foreign key (NUM_UTILIZADOR, NUM_PROFESSOR)
references PROFESSOR (NUM_UTILIZADOR, NUM_PROFESSOR);

alter table AULA
add constraint FK_AULA_LOCALIZA_SALA foreign key (NUM_SALA)
references SALA (NUM_SALA);

alter table AVALIACAO_ESCRITA
add constraint FK_AVALIACA_AGENDA_DISCIPLI foreign key (CODIGO_DISCIPLINA)
references DISCIPLINA (CODIGO_DISCIPLINA);

alter table DISCIPLINA
```

```
add constraint FK_DISCIPLI_POSSU_CURSO foreign key (NUM_CURSO)
  references CURSO (NUM_CURSO);

alter table DISCIPLINA
  add constraint FK_DISCIPLI_RESPONSAV_PROFESSO foreign key (NUM_UTILIZADOR, NUM_PROFESSOR)
  references PROFESSOR (NUM_UTILIZADOR, NUM_PROFESSOR);

alter table FAZER
  add constraint FK_FAZER_FAZ_ALUNO foreign key (NUM_UTILIZADOR, NUM_ALUNO)
  references ALUNO (NUM_UTILIZADOR, NUM_ALUNO);

alter table FAZER
  add constraint FK_FAZER_MARCACAO_TRABALHO foreign key (CODIGO_TRABALHO)
  references TRABALHO (CODIGO_TRABALHO);

alter table FREQUENTA
  add constraint FK_FREQUENT_FREQUENTA_AULA foreign key (NUM_SALA, CODIGO_AULA)
  references AULA (NUM_SALA, CODIGO_AULA);

alter table FREQUENTA
```

```
add constraint FK_FREQUENT_FREQUENTA_ALUNO foreign key (NUM_UTILIZADOR, NUM_ALUNO)
references ALUNO (NUM_UTILIZADOR, NUM_ALUNO);

alter table FUNCIONARIOS
add constraint FK_FUNCIONA_INHERITAN_UTILIZAD foreign key (NUM_UTILIZADOR)
references UTILIZADOR_DO_ISEC (NUM_UTILIZADOR);

alter table LECIONA
add constraint FK_LECIONA_ENSINA_PROFESSO foreign key (NUM_UTILIZADOR, NUM_PROFESSOR)
references PROFESSOR (NUM_UTILIZADOR, NUM_PROFESSOR);

alter table LECIONA
add constraint FK_LECIONA_MATERIA_DISCIPLI foreign key (CODIGO_DISCIPLINA)
references DISCIPLINA (CODIGO_DISCIPLINA);

alter table PERTENCE
add constraint FK_PERTENCE_PERTENCE_FUNCIONA foreign key (NUM_UTILIZADOR, NUM_FUNCIONARIO)
references FUNCIONARIOS (NUM_UTILIZADOR, NUM_FUNCIONARIO);

alter table PERTENCE
```

```
add constraint FK_PERTENCE_PERTENCE2_CARGO foreign key (CODIGO_CARGO)
```

```
references CARGO (CODIGO_CARGO);
```

```
alter table PRIVILEGIOS
```

```
add constraint FK_PRIVILEG_CONTEM_CARGO foreign key (CODIGO_CARGO)
```

```
references CARGO (CODIGO_CARGO);
```

```
alter table PRIVILEGIOS
```

```
add constraint FK_PRIVILEG_TEM_SERVICO foreign key (CODIGO_SERVICO)
```

```
references SERVICO (CODIGO_SERVICO);
```

```
alter table PROFESSOR
```

```
add constraint FK_PROFESSO_CARGO_NO_DEPARTAM foreign key (NUM_DEPARTAMENTO)
```

```
references DEPARTAMENTO (NUM_DEPARTAMENTO);
```

```
alter table PROFESSOR
```

```
add constraint FK_PROFESSO_FAZ_PARTE_CATEGORIAI foreign key (CODIGO_CATEGORIA)
```

```
references CATEGORIA (CODIGO_CATEGORIA);
```

```
alter table PROFESSOR
```

```
add constraint FK_PROFESSO_INHERITAN_UTILIZAD foreign key (NUM_UTILIZADOR)
references UTILIZADOR_DO_ISEC (NUM_UTILIZADOR);

alter table PROPINAS
add constraint FK_PROPINAS_GESTAO_ALUNO foreign key (NUM_UTILIZADOR, NUM_ALUNO)
references ALUNO (NUM_UTILIZADOR, NUM_ALUNO);

alter table REALIZACAO
add constraint FK_REALIZAC_MARCACAO2_AVALIACA foreign key (CODIGO_TRABALHO2)
references AVALIACAO_ESCRITA (CODIGO_TRABALHO2);

alter table REALIZACAO
add constraint FK_REALIZAC_PRESENCA_ALUNO foreign key (NUM_UTILIZADOR, NUM_ALUNO)
references ALUNO (NUM_UTILIZADOR, NUM_ALUNO);

alter table SALA
add constraint FK_SALA_ESTA_DEPARTAM foreign key (NUM_DEPARTAMENTO)
references DEPARTAMENTO (NUM_DEPARTAMENTO);

alter table TRABALHO
```

```
add constraint FK_TRABALHO_ENUNCIA_DISCIPLI foreign key (CODIGO_DISCIPLINA)
references DISCIPLINA (CODIGO_DISCIPLINA);

alter table UTILIZADOR_DO_ISEC
add constraint FK_UTILIZAD_HABITA_MORADA foreign key (CODIGO_MORADA)
refe/*=====
=====*/
/* DBMS name: ORACLE Version 11g
 * Created on: 12/12/2022 05:07:37
 */=====
=====*/



alter table ACAO
drop constraint FK_ACAO_ACAO_UTILIZAD;

alter table ACAO
drop constraint FK_ACAO_ACAO2_REGISTO;

alter table ALUNO
drop constraint FK_ALUNO_INHERITAN_UTILIZAD;
```

```
alter table ATRIBUIDO
drop constraint FK_ATRIBUID_ATRIBUIDO_FUNCIONA;

alter table ATRIBUIDO
drop constraint FK_ATRIBUID_ATRIBUIDO_SERVICO;

alter table AULA
drop constraint FK_AULA_DA_PROFESSO;

alter table AULA
drop constraint FK_AULA_LOCALIZA_SALA;

alter table AVALIACAO_ESCRITA
drop constraint FK_AVALIACA_AGENDA_DISCIPLI;

alter table DISCIPLINA
drop constraint FK_DISCIPLI_POSSUI_CURSO;

alter table DISCIPLINA
drop constraint FK_DISCIPLI_RESPONSAV_PROFESSO;
```

```
alter table FAZER
drop constraint FK_FAZER_FAZ_ALUNO;

alter table FAZER
drop constraint FK_FAZER_MARCACAO_TRABALHO;

alter table FREQUENTA
drop constraint FK_FREQUENT_FREQUENTA_AULA;

alter table FREQUENTA
drop constraint FK_FREQUENT_FREQUENTA_ALUNO;

alter table FUNCIONARIOS
drop constraint FK_FUNCIONA_INHERITAN_UTILIZAD;

alter table LECIONA
drop constraint FK_LECIONA_ENSINA_PROFESSO;

alter table LECIONA
```

```
drop constraint FK_LECTURA_MATERIA_DISCIPLI;
```

```
alter table PERTENCE
```

```
drop constraint FK_PERTENCE_PERTENCE_FUNCIONA;
```

```
alter table PERTENCE
```

```
drop constraint FK_PERTENCE_PERTENCE2_CARGO;
```

```
alter table PRIVILEGIOS
```

```
drop constraint FK_PRIVILEG_CONTEM_CARGO;
```

```
alter table PRIVILEGIOS
```

```
drop constraint FK_PRIVILEG_TEM_SERVICO;
```

```
alter table PROFESSOR
```

```
drop constraint FK_PROFESSO_CARGO_NO_DEPARTAM;
```

```
alter table PROFESSOR
```

```
drop constraint FK_PROFESSO_FAZ_PARTE_CATEGORI;
```

```
alter table PROFESSOR
drop constraint FK_PROFESSO_INHERITAN_UTILIZAD;

alter table PROPINAS
drop constraint FK_PROPINAS_GESTAO_ALUNO;

alter table REALIZACAO
drop constraint FK_REALIZAC_MARCACAO2_AVALIACA;

alter table REALIZACAO
drop constraint FK_REALIZAC_PRESENCA_ALUNO;

alter table SALA
drop constraint FK_SALA_ESTA_DEPARTAM;

alter table TRABALHO
drop constraint FK_TRABALHO_ENUNCIA_DISCIPLI;

alter table UTILIZADOR_DO_ISEC
drop constraint FK_UTILIZAD_HABITA_MORADA;
```

```
drop index ACAO2_FK;

drop index ACAO_FK;

drop table ACAO cascade constraints;

drop index INHERITANCE_1_FK;

drop table ALUNO cascade constraints;

drop index ATRIBUIDO2_FK;

drop index ATRIBUIDO_FK;

drop table ATRIBUIDO cascade constraints;

drop index LOCALIZA_SE_FK;

drop index DA_FK;
```

```
drop table AULA cascade constraints;

drop index AGENDA_FK;

drop table AVALIACAO_ESCRITA cascade constraints;

drop table CARGO cascade constraints;

drop table CATEGORIA cascade constraints;

drop table CURSO cascade constraints;

drop table DEPARTAMENTO cascade constraints;

drop index RESPONSAVEL_FK;

drop index POSSUI_FK;

drop table DISCIPLINA cascade constraints;
```

```
drop index FAZ_FK;

drop index MARCACAO_FK;

drop table FAZER cascade constraints;

drop index FREQUENTA2_FK;

drop table FREQUENTA cascade constraints;

drop index INHERITANCE_2_FK;

drop table FUNCIONARIOS cascade constraints;

drop index ENSINA_FK;

drop index MATERIA_FK;
```

```
drop table LECIONA cascade constraints;

drop table MORADA cascade constraints;

drop index PERTENCE2_FK;

drop table PERTENCE cascade constraints;

drop index CONTEM_FK;

drop index TEM_FK;

drop table PRIVILEGIOS cascade constraints;

drop index INHERITANCE_3_FK;

drop index FAZ_PARTE_FK;
```

```
drop index POSICAO_CARGO_NO_DEPARTAMENTO_;
```

```
drop table PROFESSOR cascade constraints;
```

```
drop index GESTAO_FK;
```

```
drop table PROPINAS cascade constraints;
```

```
drop index PRESENCA_FK;
```

```
drop index MARCACAO2_FK;
```

```
drop table REALIZACAO cascade constraints;
```

```
drop table REGISTO cascade constraints;
```

```
drop index ESTA_FK;
```

```
drop table SALA cascade constraints;
```

```
drop table SERVICO cascade constraints;

drop index ENUNCIA_FK;

drop table TRABALHO cascade constraints;

drop index HABITA_FK;

drop table UTILIZADOR_DO_ISEC cascade constraints;

/*=====
 * Table: ACAO
 * =====*/
create table ACAO

(
    NUM_UTILIZADOR      NUMBER          not null,
    ID_REGISTO          NUMBER          not null,
    constraint PK_ACAO primary key (NUM_UTILIZADOR, ID_REGISTO)
);
```

```
/*=====
 * Index: ACAO_FK
 *=====
 create index ACAO_FK on ACAO (
    NUM_UTILIZADOR ASC
);

/*=====
 * Index: ACAO2_FK
 *=====
 create index ACAO2_FK on ACAO (
    ID_REGISTO ASC
);

/*=====
 * Table: ALUNO
 *=====
 create table ALUNO
(
```

```

NUM_UTILIZADOR NUMBER      not null,
NUM_ALUNO        NUMBER      not null,
CODIGO_MORADA   NUMBER,
UTL_NOME         VARCHAR2(20),
DATA_ENTRADA    DATE,
NACIONALIDADE   VARCHAR2(20),
NOME             VARCHAR2(20),
ECTS            NUMBER,
MEDIA           NUMBER,
RAMO            VARCHAR2(20),
constraint PK_ALUNO primary key (NUM_UTILIZADOR, NUM_ALUNO)
);

/*=====
 * Index: INHERITANCE_1_FK
 *=====
 create index INHERITANCE_1_FK on ALUNO (
    NUM_UTILIZADOR ASC
);

```

```

=====
/* Table: ATRIBUIDO
*/
=====

create table ATRIBUIDO
(
    (
        NUM_UTILIZADOR      NUMBER          not null,
        NUM_FUNCIONARIO     NUMBER          not null,
        CODIGO_SERVICIO     NUMBER          not null,
        constraint PK_ATRIBUIDO primary key (NUM_UTILIZADOR, NUM_FUNCIONARIO, CODIGO_SERVICIO)
    );
);

=====
/* Index: ATRIBUIDO_FK
*/
=====

create index ATRIBUIDO_FK on ATRIBUIDO (
    NUM_UTILIZADOR ASC,
    NUM_FUNCIONARIO ASC
);
=====

/*

```

```

/* Index: ATRIBUIDO2_FK
 */
create index ATRIBUIDO2_FK on ATRIBUIDO (
    CODIGO_SERVICO ASC
);

/*=====
 * Table: AULA
 *=====
 */
create table AULA
(
    NUM_SALA          NUMBER      not null,
    CODIGO_AULA       NUMBER      not null,
    NUM_UTILIZADOR    NUMBER      not null,
    NUM_PROFESSOR     NUMBER      not null,
    DURACAO           DATE,
    HORARIO           DATE,
    constraint PK_AULA primary key (NUM_SALA, CODIGO_AULA)
);

```

```
=====
/* Index: DA_FK */
/*=====
create index DA_FK on AULA (
    NUM_UTILIZADOR ASC,
    NUM_PROFESSOR ASC
);

=====
/* Index: LOCALIZA_SE_FK */
/*=====
create index LOCALIZA_SE_FK on AULA (
    NUM_SALA ASC
);

=====
/* Table: AVALIACAO_ESCRITA */
/*=====
create table AVALIACAO_ESCRITA
(

```

```

CÓDIGO_TRABALHO2 NUMBER(10)          not null,
CÓDIGO_DISCIPLINA NUMBER,
DATA      DATE,
COTACAO   NUMBER,
DURACAO   DATE,
constraint PK_AVALIACAO_ESCRITA primary key (CÓDIGO_TRABALHO2)
);

/*=====
 * Index: AGENDA_FK
 *=====
 create index AGENDA_FK on AVALIACAO_ESCRITA(
    CÓDIGO_DISCIPLINA ASC
);

/*=====
 * Table: CARGO
 *=====
 create table CARGO
(

```

```
CODIGO_CARGO NUMBER      not null,  
NOME      VARCHAR2(20),  
constraint PK_CARGO primary key (CODIGO_CARGO)  
);
```

```
/*=====*/  
/* Table: CATEGORIA */  
/*=====*/  
create table CATEGORIA  
(  
    CODIGO_CATEGORIA NUMBER      not null,  
    NOME      VARCHAR2(20),  
    FUNCAO     CLOB,  
constraint PK_CATEGORIA primary key (CODIGO_CATEGORIA)  
);  
/*=====*/  
/* Table: CURSO */  
/*=====*/  
create table CURSO
```

```

(
    NUM_CURSO      NUMBER      not null,
    NOME           VARCHAR2(20),
    NUM_DISCIPLINAS NUMBER,
    NUM_ECTS       NUMBER,
    NUM_ANOS       NUMBER,
    constraint PK_CURSO primary key (NUM_CURSO)
);

/*=====
 * Table: DEPARTAMENTO
 *=====
 create table DEPARTAMENTO

(
    NUM_DEPARTAMENTO NUMBER      not null,
    NOME           VARCHAR2(20),
    NUM_ANDARES   NUMBER,
    constraint PK_DEPARTAMENTO primary key (NUM_DEPARTAMENTO)
);

```

```
/*=====
 * Table: DISCIPLINA
 */
/*=====*/
create table DISCIPLINA
(
    CODIGO_DISCIPLINA NUMBER          not null,
    NUM_UTILIZADOR   NUMBER,
    NUM_PROFESSOR    NUMBER,
    NUM_CURSO        NUMBER          not null,
    NOME             VARCHAR2(20),
    SEMESTRE         NUMBER,
    NUM_ECTS         NUMBER,
    constraint PK_DISCIPLINA primary key (CODIGO_DISCIPLINA)
);

/*=====
 * Index: POSSUI_FK
 */
/*=====*/
create index POSSUI_FK on DISCIPLINA(
    NUM_CURSO ASC
```

```

);

/*=====
/* Index: RESPONSAVEL_FK
 */

/*=====
create index RESPONSAVEL_FK on DISCIPLINA (
    NUM_UTILIZADOR ASC,
    NUM_PROFESSOR ASC
);

/*=====
/* Table: FAZER
 */

/*=====
create table FAZER
(
    CODIGO_TRABALHO NUMBER(10)      not null,
    NUM_UTILIZADOR NUMBER          not null,
    NUM_ALUNO      NUMBER          not null,
    DATA_ENTREGA   DATE,
    REALIZACAO    SMALLINT       not null,
);

```

```

constraint PK_FAZER primary key (CODIGO_TRABALHO, NUM_UTILIZADOR, NUM_ALUNO)
);

/*=====
 * Index: MARCACAO_FK
 */

/*=====*/
create index MARCACAO_FK on FAZER (
    CODIGO_TRABALHO ASC
);

/*=====
 * Index: FAZ_FK
 */

/*=====*/
create index FAZ_FK on FAZER (
    NUM_UTILIZADOR ASC,
    NUM_ALUNO ASC
);

/*=====*/
/* Table: FREQUENTA
 */

```

```

=====
/*=====
create table FREQUENTA
(
    (
        NUM_SALA      NUMBER      not null,
        CODIGO_AULA   NUMBER      not null,
        NUM_UTILIZADOR NUMBER      not null,
        NUM_ALUNO     NUMBER      not null,
    );
    constraint PK_FREQUENTA primary key (NUM_SALA, CODIGO_AULA, NUM_UTILIZADOR, NUM_ALUNO)
);

=====
/*=====
/* Index: FREQUENTA_FK
*/
=====
create index FREQUENTA_FK on FREQUENTA(
    NUM_SALA ASC,
    CODIGO_AULA ASC
);
=====
/*=====
/* Index: FREQUENTA2_FK
*/

```

```

/*
=====
create index FREQUENTA2_FK on FREQUENTA(
    NUM_UTILIZADOR ASC,
    NUM_ALUNO ASC
);

/*
=====
/* Table: FUNCIONARIOS
*/
/*
=====
create table FUNCIONARIOS

(
    NUM_UTILIZADOR NUMBER          not null,
    NUM_FUNCIONARIO NUMBER          not null,
    CODIGO_MORADA NUMBER,
    UTI_NOME        VARCHAR2(20),
    DATA_ENTRADA   DATE,
    NACIONALIDADE VARCHAR2(20),
    NOME           VARCHAR2(20),
    constraint PK_FUNCIONARIOS primary key (NUM_UTILIZADOR, NUM_FUNCIONARIO)
);

```

```
/*=====
 * Index: INHERITANCE_2_FK
 *=====
 create index INHERITANCE_2_FK on FUNCIONARIOS (
    NUM_UTILIZADOR ASC
);

/*=====
 * Table: LECIONA
 *=====
 create table LECIONA

(
    CODIGO_DISCIPLINA NUMBER          not null,
    NUM_UTILIZADOR   NUMBER,
    NUM_PROFESSOR    NUMBER,
    ANO_LETIVO       NUMBER          not null
);
/*=====
```

```
/* Index: MATERIA_FK
 */
create index MATERIA_FK on LECIONA (
    CODIGO_DISCIPLINA_ASC
);

/*=====
 * Index: ENSINA_FK
 */
create index ENSINA_FK on LECIONA (
    NUM_UTILIZADOR_ASC,
    NUM_PROFESSOR_ASC
);

/*=====
 * Table: MORADA
 */
create table MORADA
(
    CODIGO_MORADA      NUMBER      not null,

```

```

PAIS          VARCHAR2(20),
CIDADE        VARCHAR2(20),
CODIGO_POSTAL  VARCHAR2(8),
RUA           CLOB,
constraint PK_MORADA primary key (CODIGO_MORADA)
);

/*=====
 * Table: PERTENCE
 *=====
 create table PERTENCE

(
  NUM_UTILIZADOR NUMBER      not null,
  NUM FUNCIONARIO NUMBER      not null,
  CODIGO_CARGO   NUMBER      not null,
constraint PK_PERTENCE primary key (NUM_UTILIZADOR, NUM_FUNCIONARIO, CODIGO_CARGO)
);

/*=====
 * Index: PERTENCE_FK
 *=====

```

```
=====
/*=====
create index PERTENCE_FK on PERTENCE (
    NUM_UTILIZADOR ASC,
    NUM_FUNCIONARIO ASC
);

=====*/
/* Index: PERTENCE2_FK */
=====*/
create index PERTENCE2_FK on PERTENCE (
    CODIGO_CARGO ASC
);

=====*/
/* Table: PRIVILEGIOS */
=====*/
create table PRIVILEGIOS
(
    ID          NUMBER      not null,
    CODIGO_CARGO  NUMBER,
```

```
CODIGO_SERVICO  NUMBER,
ID_TABELA      NUMBER,
EDITAR        SMALLINT,
VER          SMALLINT,
constraint PK_PRIVILEGIOS primary key (ID)
);

/*=====
 * Index: TEM_FK
 *=====
create index TEM_FK on PRIVILEGIOS (
    CODIGO_SERVICO ASC
);

/*=====
 * Index: CONTEM_FK
 *=====
create index CONTEM_FK on PRIVILEGIOS (
    CODIGO_CARGO ASC
);
```

```

/*
 * Table: PROFESSOR
 */
create table PROFESSOR
(
    NUM_UTILIZADOR NUMBER not null,
    NUM_PROFESSOR NUMBER not null,
    NUM_DEPARTAMENTO NUMBER not null,
    CODIGO_CATEGORIA NUMBER not null,
    CODIGO_MORADA NUMBER,
    UTI_NOME VARCHAR2(20),
    DATA_ENTRADA DATE,
    NACIONALIDADE VARCHAR2(20),
    NOME VARCHAR2(20),
constraint PK_PROFESSOR primary key (NUM_UTILIZADOR, NUM_PROFESSOR)
);
/*
 * Index: POSICAO_CARGO_NO_DEPARTAMENTO_
 */

```

```
=====
/*=====
create index POSICAO_CARGO_NO_DEPARTAMENTO_on PROFESSOR (
    NUM_DEPARTAMENTO ASC
);

=====*/
/*=====
/* Index: FAZ_PARTE_FK
   */
=====*/
create index FAZ_PARTE_FK on PROFESSOR (
    CODIGO_CATEGORIA ASC
);

=====*/
/*=====
/* Index: INHERITANCE_3_FK
   */
=====*/
create index INHERITANCE_3_FK on PROFESSOR (
    NUM_UTILIZADOR ASC
);

=====*/
```

```

/* Table: PROPINAS
 */
-----*/
create table PROPINAS
(
    (
        CODIGO_PROPINA      NUMBER          not null,
        NUM_UTILIZADOR      NUMBER,
        NUM_ALUNO            NUMBER,
        MONTANTE             NUMBER(8,2),
        DATA_LIMITE          DATE,
        VENCIMENTO           SMALLINT,
        REFERENCIA           NUMBER,
        ENTIDADE              NUMBER,
        ESTADO_PAGAMENTO     SMALLINT,
        constraint PK_PROPINAS primary key (CODIGO_PROPINA)
    );
-----*/
/* Index: GESTAO_FK
 */
-----*/
create index GESTAO_FK on PROPINAS (

```

```

NUM_UTILIZADOR ASC,
NUM_ALUNO ASC
);


$$/*=====
 * Table: REALIZACAO
 *=====
 *=====*/
create table REALIZACAO
(
    CODIGO_TRABALHO2 NUMBER(10)          not null,
    NUM_UTILIZADOR NUMBER                not null,
    NUM_ALUNO      NUMBER                not null,
    DATA_ENTREGA   DATE,
    REALIZACAO    SMALLINT              not null,
)
constraint PK_REALIZACAO primary key (CODIGO_TRABALHO2, NUM_UTILIZADOR, NUM_ALUNO)
);


$$/*=====
 * Index: MARCACAO2_FK
 *=====
 *=====*/$$$$

```

```
create index MARCACAO2_FK on REALIZACAO (
    CODIGO_TRABALHO2 ASC
);

/*=====
 * Index: PRESENCA_FK
 *=====
 create index PRESENCA_FK on REALIZACAO (
    NUM_UTILIZADOR ASC,
    NUM_ALUNO ASC
);

/*=====
 * Table: REGISTO
 *=====
 create table REGISTO
(
    ID_REGISTO      NUMBER          not null,
    TAREFA          CLOB,
    LOG             DATE,
);
```

```
constraint PK_REGISTO primary key (ID_REGISTO)
```

```
);
```

```
/*=====
 */
```

```
/* Table: SALA
 * /
```

```
/*=====*/
create table SALA
```

```
(  
    NUM_SALA      NUMBER      not null,  
    NUM_DEPARTAMENTO  NUMBER      not null,  
    NOME          VARCHAR2(20),  
    CAPACIDADE    NUMBER,  
    constraint PK_SALA primary key (NUM_SALA)  
,
```

```
/*=====*/
/* Index: ESTA_FK
 * /
```

```
/*=====*/
create index ESTA_FK on SALA(  
    NUM_DEPARTAMENTO ASC
```

```
);

/*=====
 * Table: SERVICO
 */

create table SERVICO
(
    CODIGO_SERVICO      NUMBER      not null,
    DATA                 DATE,
    TAREFA               CLOB,
    constraint PK_SERVICO primary key (CODIGO_SERVICO)
);

/*=====
 * Table: TRABALHO
 */

create table TRABALHO
(
    CODIGO_TRABALHO     NUMBER(10)   not null,
    CODIGO_DISCIPLINA   NUMBER,
```

```

        DATA      DATE,
        COTACAO   NUMBER,
constraint PK_TRABALHO primary key (CODIGO_TRABALHO)
);

/*=====
 * Index: ENUNCIA_FK
 *=====
create index ENUNCIA_FK on TRABALHO (
    CODIGO_DISCIPLINA ASC
);

/*=====
 * Table: UTILIZADOR_DO_ISEC
 *=====
create table UTILIZADOR_DO_ISEC
(
    NUM_UTILIZADOR   NUMBER      not null,
    CODIGO_MORADA   NUMBER,
    NOME            VARCHAR2(20),

```

```

DATA_ENTRADA      DATE,
NACIONALIDADE    VARCHAR2(20),
constraint PK_UTILIZADOR_DO_ISEC primary key (NUM_UTILIZADOR)
);

/*=====
 * Index: HABITA_FK
 *=====
 *=====
 * create index HABITA_FK on UTILIZADOR_DO_ISEC (
  CODIGO_MORADA ASC
);

alter table ACAO
add constraint FK_ACAO_ACAO_UTILIZAD foreign key (NUM_UTILIZADOR)
references UTILIZADOR_DO_ISEC (NUM_UTILIZADOR);

alter table ACAO
add constraint FK_ACAO_ACAO2_REGISTO foreign key (ID_REGISTO)
references REGISTO (ID_REGISTO);

```

```
alter table ALUNO
  add constraint FK_ALUNO_INHERITAN_UTILIZAD foreign key (NUM_UTILIZADOR)
    references UTILIZADOR_DO_ISEC (NUM_UTILIZADOR);

alter table ATRIBUIDO
  add constraint FK_ATRIBUID_ATRIBUIDO_FUNCIONA foreign key (NUM_UTILIZADOR, NUM_FUNCIONARIO)
    references FUNCIONARIOS (NUM_UTILIZADOR, NUM_FUNCIONARIO);

alter table ATRIBUIDO
  add constraint FK_ATRIBUID_ATRIBUIDO_SERVICO foreign key (CODIGO_SERVICO)
    references SERVICO (CODIGO_SERVICO);

alter table AULA
  add constraint FK_AULA_DA_PROFESSO foreign key (NUM_UTILIZADOR, NUM_PROFESSOR)
    references PROFESSOR (NUM_UTILIZADOR, NUM_PROFESSOR);

alter table AULA
  add constraint FK_AULA_LOCALIZA_SALA foreign key (NUM_SALA)
    references SALA (NUM_SALA);
```

```
alter table AVALIACAO_ESCRITA
add constraint FK_AVALIACA_AGENDA_DISCIPLI foreign key (CODIGO_DISCIPLINA)
references DISCIPLINA (CODIGO_DISCIPLINA);

alter table DISCIPLINA
add constraint FK_DISCIPLI_POSSEU_CURSO foreign key (NUM_CURSO)
references CURSO (NUM_CURSO);

alter table DISCIPLINA
add constraint FK_DISCIPLI_RESPONSAV_PROFESSO foreign key (NUM_UTILIZADOR, NUM_PROFESSOR)
references PROFESSOR (NUM_UTILIZADOR, NUM_PROFESSOR);

alter table FAZER
add constraint FK_FAZER_FAZ_ALUNO foreign key (NUM_UTILIZADOR, NUM_ALUNO)
references ALUNO (NUM_UTILIZADOR, NUM_ALUNO);

alter table FAZER
add constraint FK_FAZER_MARCACAO_TRABALHO foreign key (CODIGO_TRABALHO)
references TRABALHO (CODIGO_TRABALHO);
```

```

alter table FREQUENTA
add constraint FK_FREQUENT_FREQUENTA_AULA foreign key (NUM_SALA, CODIGO_AULA)
references AULA (NUM_SALA, CODIGO_AULA);

alter table FREQUENTA
add constraint FK_FREQUENT_FREQUENTA_ALUNO foreign key (NUM_UTILIZADOR, NUM_ALUNO)
references ALUNO (NUM_UTILIZADOR, NUM_ALUNO);

alter table FUNCIONARIOS
add constraint FK_FUNCIONA_INHERITAN_UTILIZAD foreign key (NUM_UTILIZADOR)
references UTILIZADOR_DO_ISEC (NUM_UTILIZADOR);

alter table LECIONA
add constraint FK_LECIONA_ENSINA_PROFESSO foreign key (NUM_UTILIZADOR, NUM_PROFESSOR)
references PROFESSOR (NUM_UTILIZADOR, NUM_PROFESSOR);

alter table LECIONA
add constraint FK_LECIONA_MATERIA_DISCIPLI foreign key (CODIGO_DISCIPLINA)
references DISCIPLINA (CODIGO_DISCIPLINA);

```

```
alter table PERTENCE
add constraint FK_PERTENCE_PERTENCE_FUNCIONA foreign key (NUM_UTILIZADOR, NUM_FUNCIONARIO)
references FUNCIONARIOS (NUM_UTILIZADOR, NUM_FUNCIONARIO);

alter table PERTENCE
add constraint FK_PERTENCE_PERTENCE2_CARGO foreign key (CODIGO_CARGO)
references CARGO (CODIGO_CARGO);

alter table PRIVILEGIOS
add constraint FK_PRIVILEG_CONTEM_CARGO foreign key (CODIGO_CARGO)
references CARGO (CODIGO_CARGO);

alter table PRIVILEGIOS
add constraint FK_PRIVILEG_TEM_SERVICO foreign key (CODIGO_SERVICO)
references SERVICO (CODIGO_SERVICO);

alter table PROFESSOR
add constraint FK_PROFESSO_CARGO_NO_DEPARTAM foreign key (NUM_DEPARTAMENTO)
references DEPARTAMENTO (NUM_DEPARTAMENTO);
```

```
alter table PROFESSOR
add constraint FK_PROFESSO_FAZ_PARTE_CATEGORIA foreign key (CODIGO_CATEGORIA)
references CATEGORIA (CODIGO_CATEGORIA);

alter table PROFESSOR
add constraint FK_PROFESSO_INHERITAN_UTILIZAD foreign key (NUM_UTILIZADOR)
references UTILIZADOR_DO_ISEC (NUM_UTILIZADOR);

alter table PROPINAS
add constraint FK_PROPINAS_GESTAO_ALUNO foreign key (NUM_UTILIZADOR, NUM_ALUNO)
references ALUNO (NUM_UTILIZADOR, NUM_ALUNO);

alter table REALIZACAO
add constraint FK_REALIZAC_MARCACAO2_AVALIACA foreign key (CODIGO_TRABALHO2)
references AVALIACAO_ESCRITA (CODIGO_TRABALHO2);

alter table REALIZACAO
add constraint FK_REALIZAC_PRESENCA_ALUNO foreign key (NUM_UTILIZADOR, NUM_ALUNO)
references ALUNO (NUM_UTILIZADOR, NUM_ALUNO);
```

```

alter table SALA
add constraint FK_SALA_ESTA_DEPARTAM foreign key (NUM_DEPARTAMENTO)
references DEPARTAMENTO (NUM_DEPARTAMENTO);

alter table TRABALHO
add constraint FK_TRABALHO_ENUNCIA_DISCIPLI foreign key (CODIGO_DISCIPLINA)
references DISCIPLINA (CODIGO_DISCIPLINA);

alter table UTILIZADOR_DO_ISEC
add constraint FK_UTILIZAD_HABITA_MORADA foreign key (CODIGO_MORADA)
references MORADA (CODIGO_MORADA);

references MORADA (CODIGO_MORADA);

```

Cap 5 - Conclusões

Com a realização deste trabalho tivemos a oportunidade de aplicar e aprofundar os nossos conhecimentos na estruturação e planificação de uma base de dados bem como ganhamos uma nova perspetiva daquilo que poderá ser uma nova estrutura, mais autónoma e optimizada da base de dados do ISEC.

Apesar de ser um trabalho que exige algumas horas dedicadas no mesmo gostamos de o realizar.

Referências Bibliográficas

- QNQ - Quadro Nacional de Qualificações: https://www.dges.gov.pt/pt/quadro_qualificacoes

Anexos

https://moodle.isec.pt/moodle/pluginfile.php/404876/mod_wiki/attachments/1036/Modelo%20Conceptual.cdm

https://moodle.isec.pt/moodle/pluginfile.php/404876/mod_wiki/attachments/1036/Modelo%20Conceptual.pdf

https://moodle.isec.pt/moodle/pluginfile.php/404876/mod_wiki/attachments/1036/Modelo%20F%C3%ADsico.pdm

https://moodle.isec.pt/moodle/pluginfile.php/404876/mod_wiki/attachments/1036/Modelo%20F%C3%ADsico.pdf