

Exercício-Programa 1: Base de dados

Entrega: 05/04/2019

Objetivos

Os objetivos do conjunto de EP's dessa disciplina são voltados para a aplicação dos conceitos estudados em sala junto ao desenvolvimento, implementação e deploy de um sistema completo baseado em um problema real.

Neste primeiro EP nos iniciaremos a implementação de um modelo persistente de gestão de grades curriculares dentro da USP, capaz de:

- Permitir a administradores a gestão de grades curriculares;
- Permitir a alunos o planejamento e o controle das disciplinas cursadas;
- Permitir o controle das disciplinas requisitadas pelas professores e dos eventuais oferecimentos dessas disciplinas;
- Permitir a gestão do sistema de módulos e trilhas que compõem diversos cursos da USP;
- Entre outras.

Para agora, faremos apenas a implementação da base de dados. Entretanto, o sistema prevê também um módulo de acesso que controlará usuários, perfis e seus respectivos acessos aos serviços oferecidos pelo Front-End, que também faz parte da base e portanto será desenvolvido.

Conhecimentos envolvidos

A solução desse EP envolve o conhecimento dos seguintes conceitos:

- Modelagem conceitual de bases de dados;
- Mapeamento de modelos conceituais para lógicos em bases de dados relacionais;
- Mapeamento de modelos lógicos para físicos utilizando o SGBD PostgreSQL.

Organização do EP

Neste EP vocês deverão trabalhar em grupo de até 3 pessoas.

Apenas um dos integrantes do grupo deverá submeter o arquivo compactado até o dia 05/04/2019 às 23:55 e este será denominado como NUSP1 a partir desse ponto.

Para a entrega, o arquivo deverá conter a seguinte estrutura:

```
EP1_NUSP1_NUSP2_NUSP3.zip
├── EP1_README.txt
├── EP1_NUSP1_REPORT.pdf
├── EP1_NUSP1_DDL.sql
└── EP1_NUSP1_DDL_CLEAN.sql
```

Onde NUSP1, NUSP2 e NUSP3 correspondem ao número USP de cada um dos integrantes, separados por um underscore. (Obs.: Nos casos dos grupos individuais ou em duplas, basta omitir os NUSP's correspondentes)

O arquivo EP1_README.txt deve conter no cabeçalho o **nome completo** e NUSP de todos os integrantes e eventuais instruções de instalação/execução/uso que eventualmente forem necessárias. (Para esse primeiro EP provavelmente não haverá necessidade de incluir qualquer instrução)

No arquivo EP1_NUSP1_REPORT.pdf¹ vocês deverão entregar um relatório formal

¹Sugerimos o uso de L^AT_EX e submissões em outro formato que não seja .pdf não serão consideradas

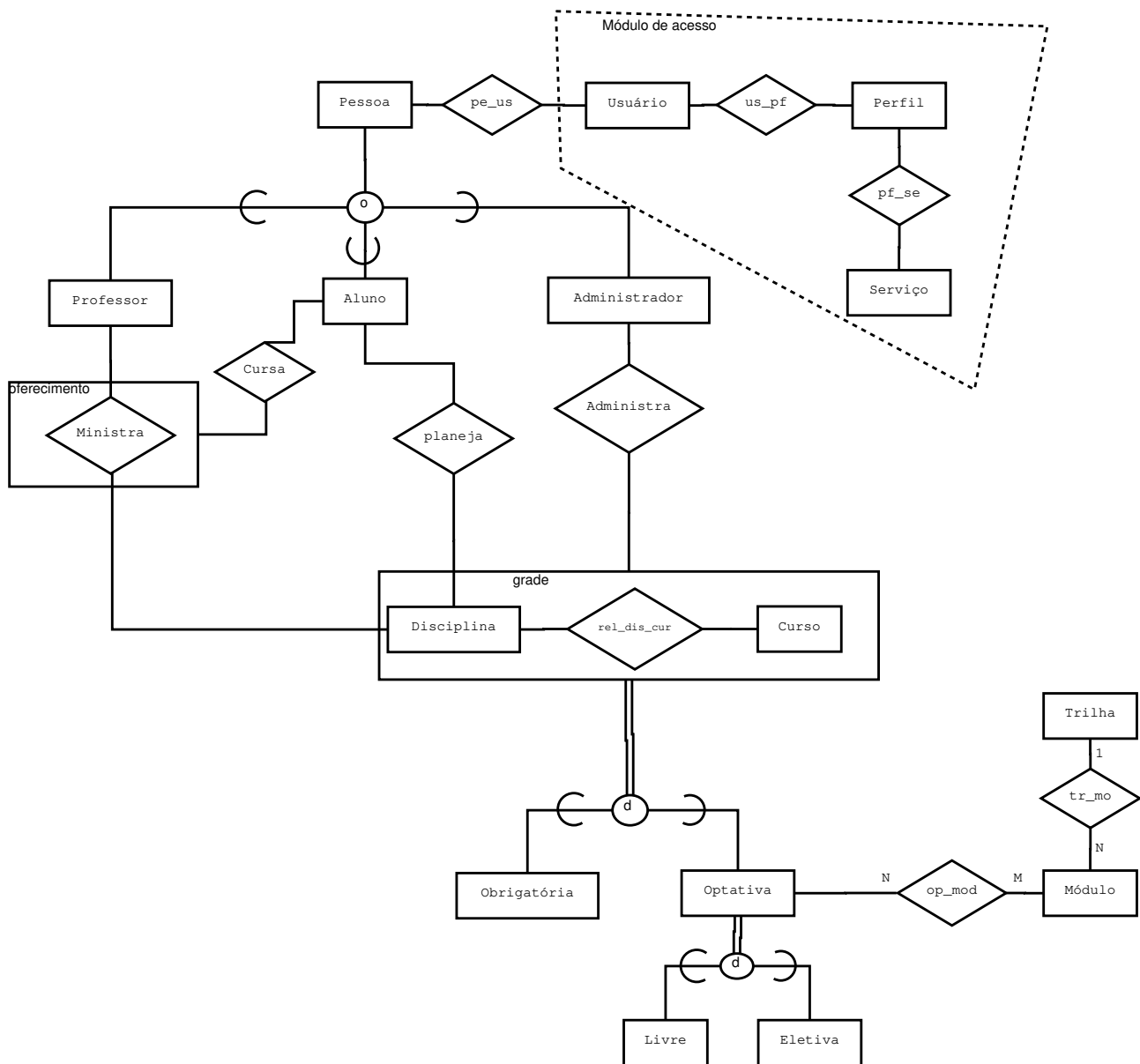


Fig. 1: Modelo conceitual simplificado que deve servir de base para o seu MER-X

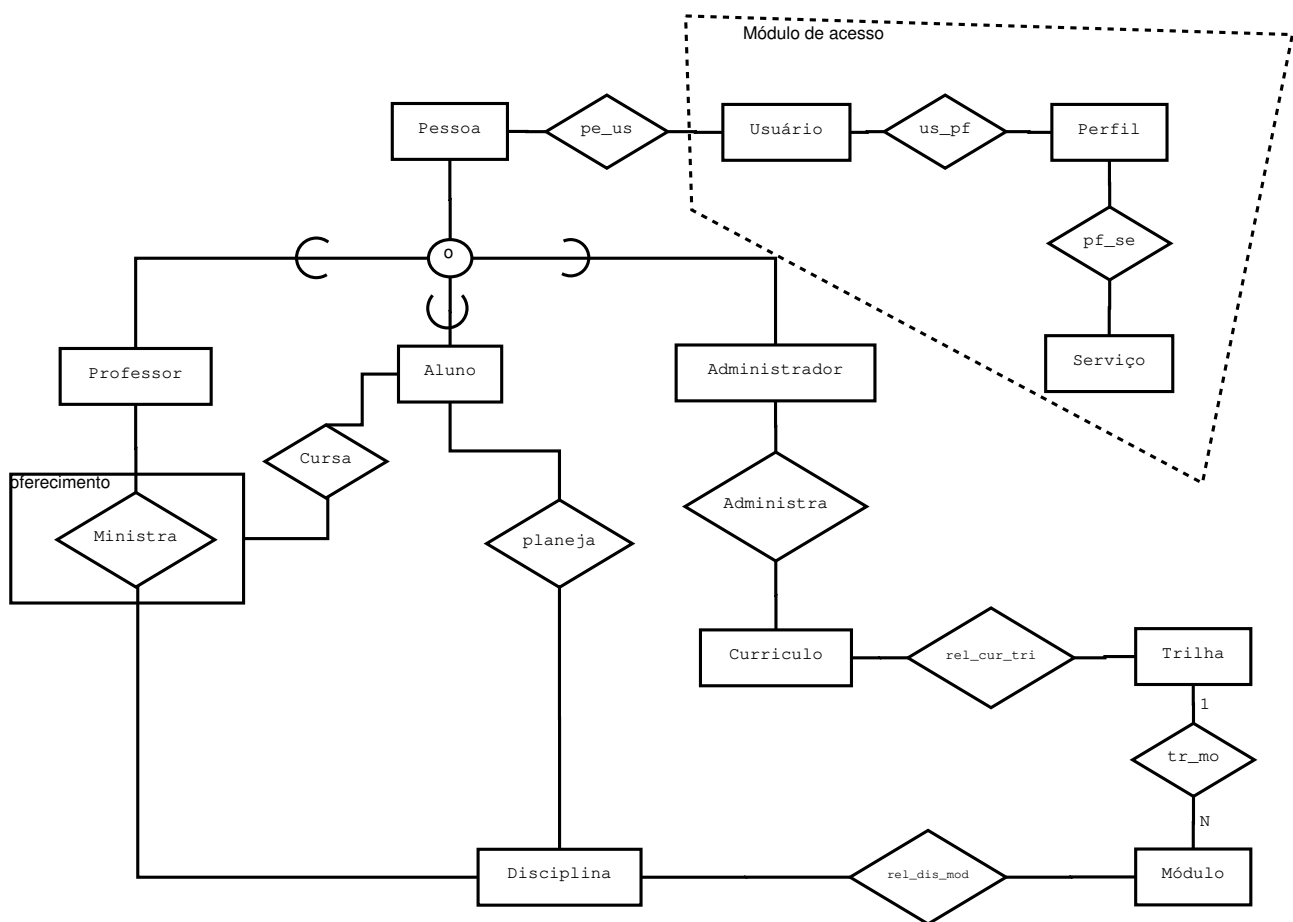


Fig. 2: Modelo conceitual simplificado versão II

contendo:

- Capa;
- Sumário;
- Complementação do modelo inicial fornecido (validação/atributos e chaves) e sua apresentação em MER-X, incluindo descrição das entidades e relacionamentos e seus respectivos atributos e chaves no modelo conceitual completo;
- Breve descrição das funcionalidades esperadas e restrições de integridade e domínio;
- Mapeamento do modelo conceitual completo para o modelo lógico;
- Complementação das restrições de integridade e restrições de domínio;
- Mapeamento do modelo lógico para o modelo físico considerando o SGBD PostgreSQL.

O arquivo *DDL.sql deve apresentar o Data Description Language do seu modelo físico e o *DDL_CLEAN.sql deve apresentar o DDL para remoção completa da base.

A seção Resumo das atividades apresenta uma descrição dos itens que compõem este EP.

Resumo das atividades

Basicamente, além da concepção do relatório em .pdf, as atividades resumem-se em:

- Entender e validar um dos modelos apresentado nas figura 1 e 2;
- Completar o modelo por meio da inclusão dos atributos de cada entidade e relacionamento, bem como das definições de chave, obtendo um diagrama Entidade-Relacionamento Estendido (DER-X);
- Mapear esse DER-X para o modelo lógico;

- Mapear o modelo lógico para o modelo físico a ser executado sobre o SGBD PostgreSQL.

Item I

O item I é autoexplicativo.

Item II

No item II, para cada entidade, relacionamento, agregação, especialização e/ou generalização, você deve incluir no relatório uma descrição geral da classe abstrata e conceber e descrever os atributos/chaves que devem fazer parte do modelo conceitual dessa classe. Abaixo apresentamos um possível exemplo de como proceder com essa apresentação.

(Exemplo)

Entidade Regular: Aluno

A entidade regular Aluno é uma entidade chave para nosso modelo e é nela que armazenaremos todos os dados das instâncias de cada um dos aluno da USP presentes na nossa base de dados.

Segundo nosso modelo, cada instância de Aluno será composta por seu número USP, que será utilizado como chave primária, seu CPF que será utilizado como chave secundária, seu nome que não admite valores do tipo NULL, seu endereço que será um atributo composto de logradouro, rua, ..., seu ... (continuar listando detalhadamente cada um dos atributos).

O listing abaixo traz um exemplo da estrutura da entidade:

(Exemplo) A entidade regular Aluno

```
Aluno
-Número USP *Chave primária
-CPF **Chave secundária
-Nome
-Idade
-Sexo
-Curso
-Endereço
:Continuar listando
```

Enquanto que a figura 3 apresenta a representação em MER-X da entidade descrita.

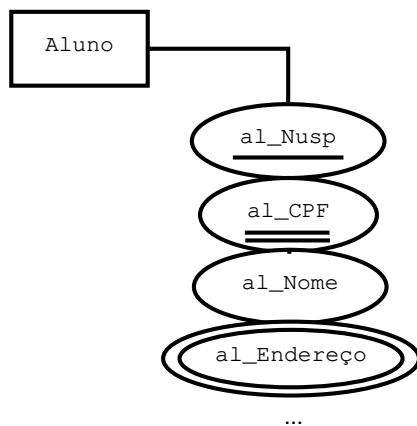


Fig. 3: Aluno em formato MER-X (Obs.: **In-completo**)

Ainda no item II, você deve fazer também uma descrição textual das restrições de integridade e domínio. Abaixo um possível exemplo de como fazer isso para a entidade aluno.

(Exemplo)

Restrições Entidade Regular: Aluno

A entidade Aluno possui as seguintes restrições de integridade/domínio:

- O atributo `al_Nusp` é composto por uma sequência de 6 a 9 dígitos inteiros (Obs.: Não sei se existe NUSP com 6 dígitos e estou supondo que o sistema já prevê a extensão para Nusp com 9 dígitos, tudo isso deve ser descrito) e por ser chave não deve permitir a presença de entradas do tipo NULL;
- O atributo `al_CPF` é composto por uma sequência de 11 algarismos e deve ser constituída de uma sequência válida segundo às regras da Receita Federal (listar as regras de congruência, dígito verificador de estado, ...), além disso, por ser chave secundária esse atributo deve ser do tipo NOTNULL(NN);
- O atributo endereço é um atributo composto que será dividido em (CEP, NomeRua, NumRua, Compl, Bairro, ...),

o CEP é composto por ... e deve ser do tipo NN, se possível, implementaremos um verificador de endereço via CEP, caso contrário, o nome da rua será uma string variável de tamanho X, o Compl pode ser do tipo NULL nos casos onde ele não exista, ...;

- O atributo sexo deve ser um char presente no conjunto (M, F, N) onde M representa Masculino, F feminino e N não declarado;
- : (Obs.: Continuar listando todas as restrições de domínio)

O listing traz um exemplo de possíveis tuplas à serem inseridas em aluno:

Exemplos de entrada para as Entidades

```
Aluno: (al_Nusp, al_CPF,
        al_Nome, al_idade, al_sexo,
        ...)
-77777777; 003.939.708-41; Juca
Medonho; 22; M; ...
-77778888; 076.713.684-58; Célia
Nefasta; 18; N; ...
```

Ao terminar vocês **devem** apresentar no relatório o DER-X completo.

Finalmente, embora não faça parte diretamente do item II, vocês ainda precisam fazer uma breve descrição das funcionalidades previstas para o seu sistema. Um possível exemplo:

(Exemplo) Funcionalidades esperadas

Nosso projeto prevê as seguintes funcionalidades:

- Permitir ao aluno planejar um cronograma de disciplinas segundo uma trilha escolhida;
- Permitir ao administrador recuperar os dados dos alunos e suas notas nas disciplinas cursadas;
- Permitir ao administrador verificar o número de alunos que completaram determinada trilha;
- :

Item III

Para o item III vocês devem apresentar o mapeamento do modelo conceitual em MER-X concebido no item II para o modelo lógico.

Uma sugestão é usar o tikzpicture, o mapeamento apresentado a seguir é um exemplo que foi gerado em L^AT_EX com inserção direta do código tikz no arquivo .tex.

ALUNO

<u>a_cpf</u>	<u>NUSP</u>	Nome	p_cpf
--------------	-------------	------	-------

IC

<u>ID</u>	titulo
-----------	--------

PROFESSOR

<u>p_cpf</u>	Nome
--------------	------

⋮

OBRIGATÓRIA

<u>ID</u>	<u>sigla</u>	ementa	peso
-----------	--------------	--------	------

Item IV

Para esse item, você deverá fazer o mapeamento do modelo lógico para o modelo físico escrevendo dois scripts SQL (*EP1_NUSP1_DDL.sql* e *EP1_NUSP1_DDL_CLEAN.sql*), sendo o primeiro para a criação e configuração da base de dados e o segundo para limpeza/remoção completa do banco.

Nesta primeira parte do projeto, não há necessidade de integrar os scripts SQL a qualquer tipo de interface em Python, ou seja, seu script deve apenas ser capaz de criar as relações no SGBD PostgreSQL e toda a análise será feita via interface direta com o banco (por exemplo, via *psql*).

Por fim, seus scripts podem estar inclusos no corpo do relatório, mas **devem** ser entregues em arquivos separados denominados *EP1_NUSP1_DDL.sql* e *EP1_NUSP1_DDL_CLEAN.sql*.

Instruções para entrega

Você deve submeter via PACA os arquivos descritos na organização, compactados em formato .zip e seguindo as especificações de estrutura e nomenclatura previamente estabelecidas, até às 23:55 do dia 05/04/2019.

Para evitar que seu EP seja zerado, certifique-se que o arquivo foi submetido sem problemas (baixando, descompactando e testando todos os arquivos), que o relatório está em formato PDF e que o script SQL foi escrito para o PostgreSQL.

Além disso, não deixe de preencher o cabeçalho contendo o nome completo e NUSP de todos os integrantes e atentem para as questões de integridade acadêmica, ou seja, caso você utilize diretamente, ou se baseie fortemente em fontes que não sejam as fornecidas em sala, não se esqueça de referenciá-las no relatório. Lembrem-se que casos de plágio serão tratados com o rigor acadêmico esperado.

Avaliação

Seu EP será avaliado pela qualidade do relatório, dos mapeamentos e dos scripts SQL. Sua nota final será composta por:

EP Parte 1 - Nota máxima 10.0
├ Relatário PDF (8.0)
│ ├ Itens I e II (Conceitual e discussões) - 5.0
│ ├ Item III (Lógico) - 3.0
│ └ Estrutura do texto e qualidade das discussões bônus de até 0.5
├ Item IV (Físico) (2.0)
│ ├ DDL.sql - 1.5
│ ├ DDL_CLEAN.sql - 0.5
│ └ Bônus de até 1.0 para alunos que implementarem também o DML

Não deixe para a última hora e bom trabalho!

Referências

- [1] Jacques Crémer. A very minimal introduction to tikz. <https://cremeronline.com/LaTeX/minimaltikz.pdf>. Accessed: 20/03/19.

- [2] Ramez Elmasri and Shamkant B. Navathe. *Database Systems, 7th Ed.* Pearson, 2015.
- [3] Décio Lauro Soares and João Eduardo Ferreira. Slides da aula de revisão de mapeamento (2019). www.paca.ime.usp.br, 2019. Accessed: 20/03/19.
- [4] Décio Lauro Soares, Eduardo Dias Filho, and Bruno Padilha. Tutorial python e postgresql. www.paca.ime.usp.br. Accessed: 20/03/19.
- [5] Osvaldo Kotaro Takai, Isabel Cristina Italiano, and João Eduardo Ferreira. Introdução à banco de dados. www.ime.usp.br/~jef/apostila.pdf, 2005. Accessed: 20/03/19.
- [6] Dia volunteer development group. Dia diagram editor. <http://dia-installer.de/>. Accessed: 20/03/19.