

## MAC0350 - EP2

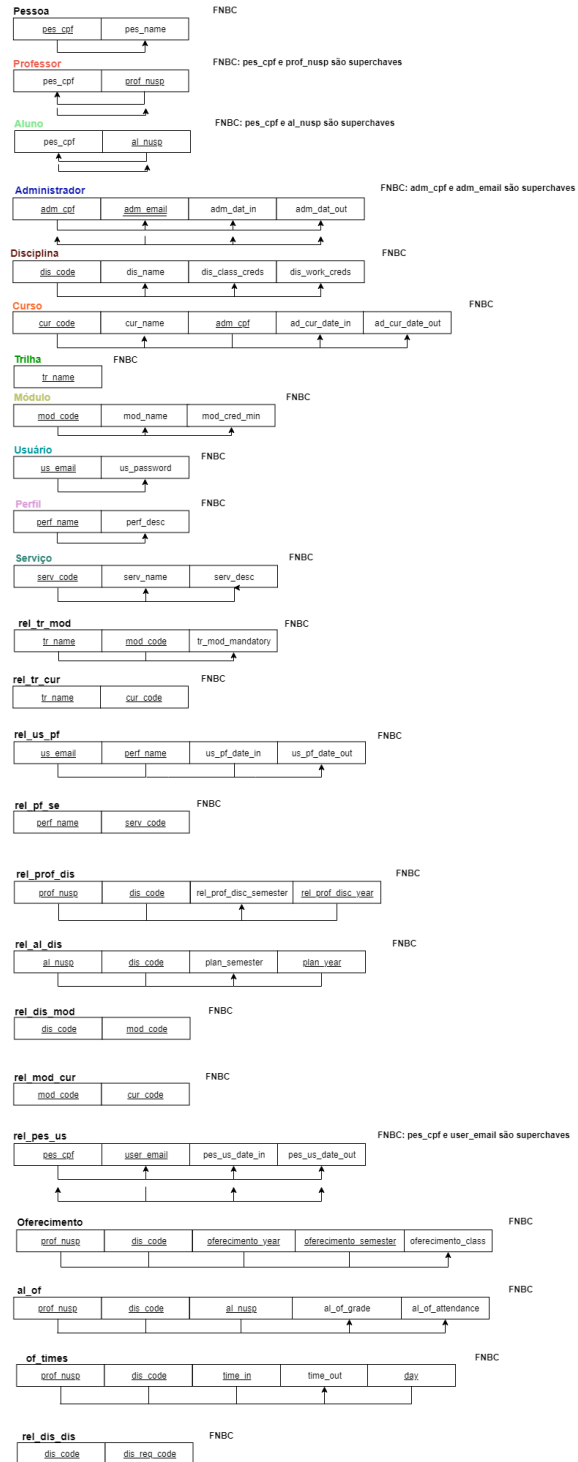
André Ferrari Moukarzel  
9298169

Arthur Vieira Barbosa  
6482041

Gabriel Sarti Massukado  
10284177

Matheus Lima Cunha  
10297755

# Compressão do Modelo



Na imagem acima temos os diagramas que indicam as dependências funcionais das relações de nosso banco de dados, junto com sua forma normal escrita ao lado, e algumas justificativas para as dependências que não atendem a Forma Normal de Boyce-Codd de forma trivial.

## Mudanças no Modelo

Além de manter no nosso modelo apenas os atributos que julgamos estritamente necessários para o funcionamento da base (removendo os que consideramos não-essenciais, como sexo da Pessoa e departamento do Professor), também fizemos algumas modificações no modelo em si.

Compreendendo melhor como funcionarão os Serviços no nosso banco de dados, percebemos que os atributos de nossa classe Serviços eram desnecessariamente complicados. Removemos a tabela de relações `serv_tables` e os `serv_per_create`, `read`, `update`, `delete`, pois julgamos que ter o nome das funções relacionadas a cada Serviço em `ser_name` é o suficiente para definir as suas funcionalidades.

Também corrigimos as chaves de várias relações, e criamos a relação `rel_dis_dis`, onde determinamos os pre-requisitos das disciplinas em uma relação N:M com os códigos das disciplinas (`dis_code`) e os códigos das disciplinas necessárias para cursá-las (`dis_req_code`).

## CREATE

Nosso `DML_CREATE_GROUP` possui uma função de criação para cada entidade e relação do nosso modelo. No caso de entidades, sua criação retorna o seu id artificial, enquanto no caso de relações, que não possuem tal id, retornamos as chaves da relação.

Para isso, também criamos várias tuplas com os sufixos *key* para retornar as chaves compostas de relações. Estas tuplas também são utilizadas nas operações de `DELETE`, `UPDATE` e `RETRIEVAL`.

Em uma função `CREATE` são passados como argumentos todos os atributos da instância a ser criada.

## DELETE

Nosso `DML_DELETE_GROUP` possui uma função de deleção para cada entidade e relação do nosso modelo. No caso de entidades, sua deleção retorna o seu id artificial, enquanto no caso de relações, que não possuem tal id, retornamos as chaves da relação.

Em uma função `DELETE`, é passada como argumento a chave primária da instância a ser deletada.

## UPDATE

Nosso DML\_UPDATE\_GROUP possui uma função de atualização para cada atributo de cada entidade e relação do nosso modelo. No caso de entidades, sua atualização retorna o seu id artificial, enquanto no caso de relações, que não possuem tal id, retornamos as chaves da relação.

Em uma função UPDATE, são passados a chave primária da instância a ser atualizada e o novo valor do atributo a ser atualizado.

Realisticamente, não deve ser necessário criar uma função de atualização para cada atributo de cada entidade, mas como não sabemos exatamente como seria utilizado o sistema, preferimos pecar por excesso.

## RETRIEVAL

Nosso DML\_RETRIEVAL\_GROUP possui funções "triviais" de retorno, todas com o prefixo *select* seguido do nome da tabela, onde a chave primária é um argumento. Todas estas funções triviais retornam o atributo referido da instância que possui a chave correspondente à enviada como argumento.

Criamos também funções não triviais no RETRIEVAL, como uma função que, dado um Oferecimento, retorna todos os Alunos nele inscritos, ou uma função que dado um Oferecimento e um valor X retorna todos os Alunos nele inscritos que tiveram nota maior que X. O nome deste tipo de função é determinado por sua funcionalidade, e fizemos comentários em cada uma delas com o fim de deixar mais claro seu objetivo.