

REFATORAÇÃO EM BANCOS DE DADOS PARTE 4

Sérgio Mergen

Processos de Refatoração

- Nessa aula veremos os seguintes processos
 - Fusão de colunas
 - Divisão de colunas
 - Deslocamento de Coluna

Fusão de Colunas

- Os endereços de email são divididos em duas partes (usuario e servidor).
- Essa divisão tinha o objetivo de permitir filtros mais eficientes sobre a coluna servidor
- No entanto, esse tipo de filtro nunca é utilizado

Func	
*idFunc	int
Nome	char(30)
Usuario_email	char(10)
Servidor_email	char(20)

Fusão de Colunas

- Os endereços de email são divididos em duas partes (usuario e servidor).
- Essa divisão tinha o objetivo de permitir filtros mais eficientes sobre a coluna servidor
- No entanto, esse tipo de filtro nunca é utilizado
- Desse modo, decidiu-se fundir essas duas colunas

Func	
*idFunc	int
Nome	char(30)
email	char(30)

Fusão de Colunas

- Script para fusão de colunas
 - Passo 1: Combinar os valores separados em uma só coluna
 - Passo 2: Remover as colunas obsoletas
- Obs. A fusão pode tornar irrecuperáveis os valores originais

Fusão de Colunas

- Script para fusão de colunas
 - Passo 1: Combinar os valores separados em uma só coluna

- Ex.

//Inicialmente, deve-se garantir que os campos alvos não possuam valores nulos

```
UPDATE func SET usuario_email = '' WHERE usuario_email IS NULL
```

```
UPDATE func SET servidor_email = '' WHERE servidor_email IS NULL
```

//atualização do valor mesclado (mySQL)

```
UPDATE func SET usuario_email =  
    concat (usuario_email, '@', servidor_email)
```

- Obs. Concat (null, 'qualquer coisa') = null

Fusão de Colunas

- Script para fusão de colunas
 - Passo 2: Remover as colunas obsoletas
- Ex.
`ALTER TABLE func DROP servidor_email`
- Como foi reaproveitada uma coluna existente
 - deve-se remover apenas `servidor_email`
- Se nenhuma das colunas fosse aproveitada
 - o script teria que ser precedido pela adição de uma coluna para receber o resultado da fusão
 - E as duas colunas originais teriam que ser removidas

Fusão de Colunas

- A combinação dos valores no passo 1 pode ocorrer de diversas formas, dependendo dos tipos das colunas
- Ex.
 - Concatenação de valores textuais
 - Cálculos aritméticos sobre valores numéricos
 - ...
- O exemplo demonstrou a concatenação de dois valores textuais
 - Outras formas de fusão podem exigir soluções diferentes
- No exemplo, é possível recuperar os valores originais
 - Como veremos no processo de refatoração a seguir

Processos de Refatoração

- Nessa aula veremos os seguintes processos
 - Fusão de colunas
 - Divisão de colunas
 - Deslocamento de Coluna

Divisão de colunas

- Os endereços de email são guardados em um único campo
- Essa forma de armazenamento torna mais cara as consultas que filtrem pelo servidor de email.

Func	
*idFunc	int
Nome	char(30)
email	char(30)

Divisão de colunas

- Os endereços de email são guardados em um único campo
- Essa forma de armazenamento torna mais cara as consultas que filtrem pelo servidor de email.
- Desse modo, decidiu-se dividir a coluna em duas

Func	
*idFunc	int
Nome	char(30)
Usuario_email	char(10)
Servidor_email	char(20)

Divisão de colunas

- Script para divisão de colunas
 - Passo 1: Criar as novas colunas
 - Passo 2: Dividir o valor no número de valores desejados
 - Passo 3: Remover as colunas obsoletas (se for o caso)
- Obs. Nem sempre é possível aplicar a divisão

Divisão de colunas

- Script para divisão de colunas
 - Passo 1: Criar as novas colunas
- Ex.

```
ALTER TABLE func ADD COLUMN usuario_email char(10)
```



```
ALTER TABLE func ADD COLUMN servidor_email char(20)
```
- Nesse caso decidiu-se por não reaproveitar a coluna existente (email)
 - Essa coluna deverá ser removida no final do script

Divisão de colunas

- Script para divisão de colunas
 - Passo 2: Dividir o valor no número de valores desejados
- Ex. (mySQL)
UPDATE **func** SET
 usuario_email = SUBSTRING_INDEX (**email**, '@', 1),
 servidor_email = SUBSTR (**email**, LOCATE ('@', **email**) + 1)
- As funções disponibilizadas para manipulação de strings dependem do SGBD
- Outros SGBDs podem oferecer mecanismos de split.

Divisão de colunas

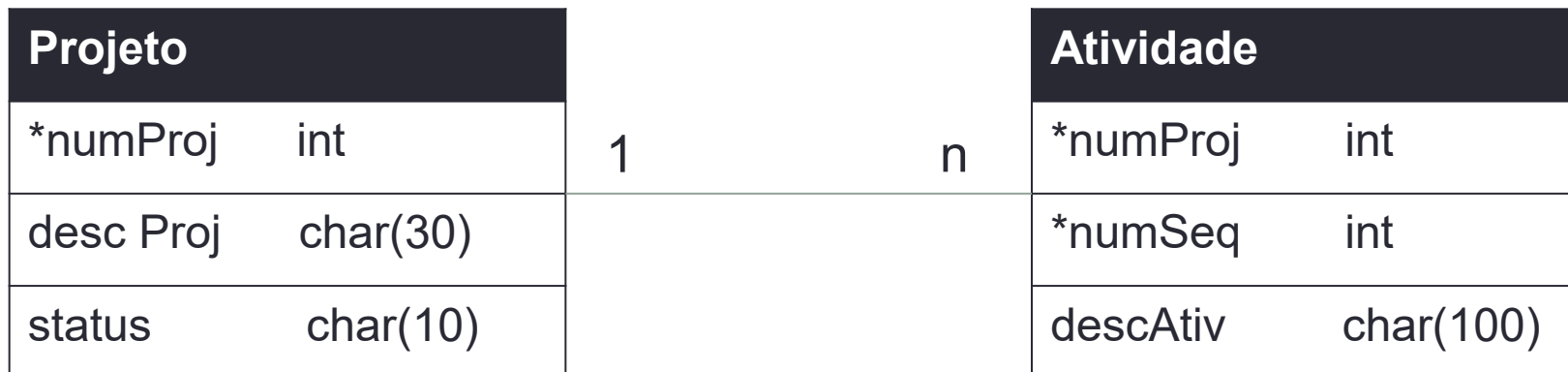
- Script para divisão de colunas
 - Passo 3: Remover as colunas obsoletas (se for o caso)
- Ex.
`ALTER TABLE func DROP email`

Processos de Refatoração

- Nessa aula veremos os seguintes processos
 - Fusão de colunas
 - Divisão de colunas
 - Deslocamento de Coluna

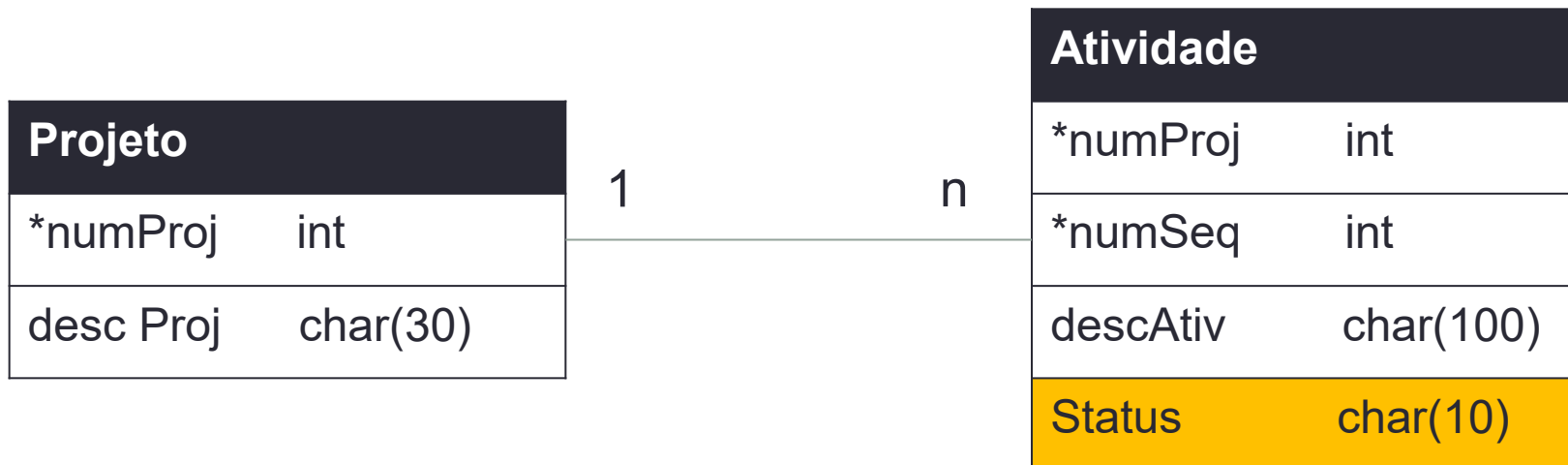
Deslocamento de Coluna

- O status de um projeto pode ser concluído ou em andamento.
- Devido a uma alteração nos requisitos, o projeto passará a ser considerado concluído quando o status de todas suas atividades for “concluído”



Deslocamento de Coluna

- O status de um projeto pode ser concluído ou em andamento.
- Devido a uma alteração nos requisitos, o projeto passará a ser considerado concluído quando o status de todas suas atividades for “concluído”
- Dessa forma, deve-se mover status de projeto para atividade



Deslocamento de Coluna

- Script para deslocamento de coluna
 - Passo 1: criação da coluna na tabela destino
 - Passo 2: alimentação da coluna na tabela destino
 - Passo 3: remoção da coluna na tabela de origem

Deslocamento de Coluna

- Script para deslocamento de coluna
 - Passo 1: criação da coluna na tabela destino
- Ex.

```
ALTER TABLE atividade ADD COLUMN status char(10);
```

Deslocamento de Coluna

- Script para deslocamento de coluna
 - Passo 2: alimentação da coluna na tabela destino
- Ex.

```
UPDATE atividade a SET status =  
    (SELECT status FROM projeto p WHERE p.idProj = a.idProj )
```

- Obs. Na migração, todas atividades recebem o mesmo status do projeto
 - Se o projeto estiver “em andamento”, todas atividades receberão o mesmo status, mesmo aquelas já concluídas
 - Como resolver isso?

Deslocamento de Coluna

- Script para deslocamento de coluna
 - Passo 2: alimentação da coluna na tabela destino
- Ex.

```
UPDATE atividade a SET status =  
    (SELECT status FROM projeto p WHERE p.idProj = a.idProj )
```

- Obs. Na migração, todas atividades recebem o mesmo status do projeto
 - Se o projeto estiver “em andamento”, todas atividades receberão o mesmo status, mesmo aquelas já concluídas
 - Como resolver isso?
 - Provavelmente o detalhamento correto não seja possível

Deslocamento de Coluna

- Script para deslocamento de coluna
 - Passo 3: remoção da coluna na tabela de origem
- Ex.

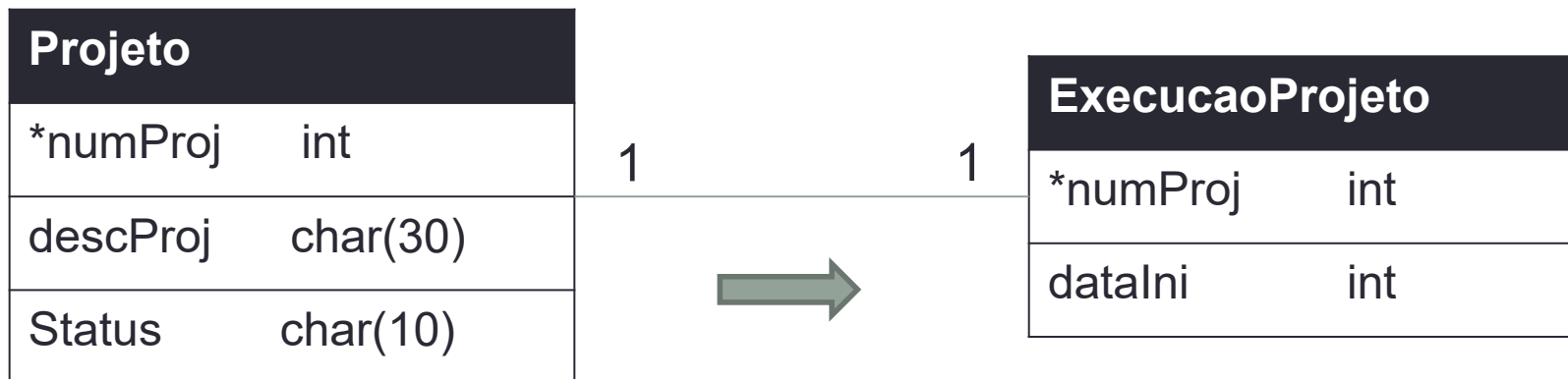
```
ALTER TABLE projeto DROP COLUMN status
```

Deslocamento de Coluna

- A complexidade do deslocamento depende da cardinalidade do relacionamento das tabelas envolvidas
- Supondo deslocamento da esquerda para a direita, existem três possibilidades
 - Quando a cardinalidade é 1-1
 - Quando a cardinalidade é 1-n
 - Quando a cardinalidade é n-1

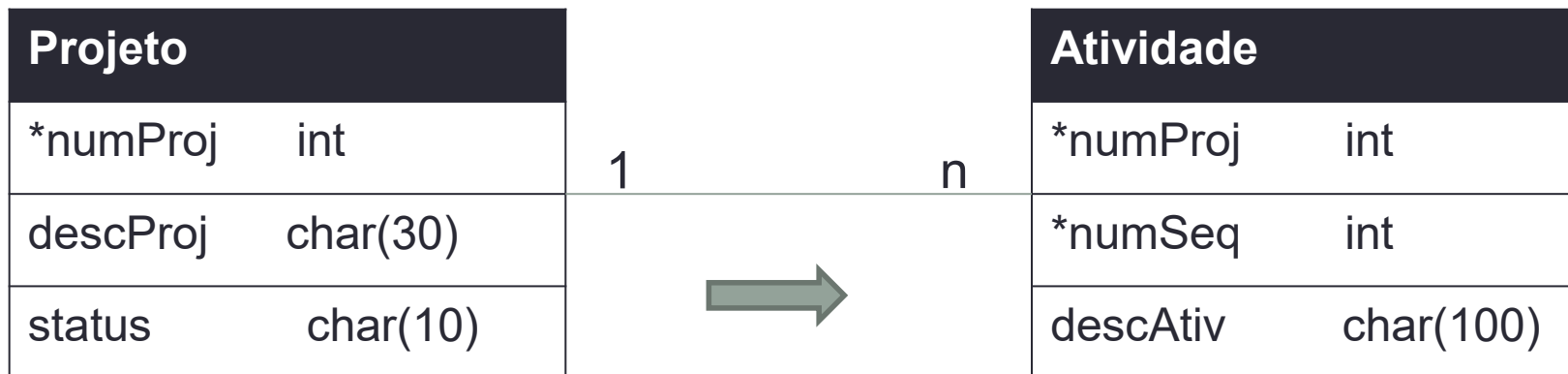
Deslocamento de Coluna

- Quando a cardinalidade é 1-1
 - A alimentação é trivial (basta copiar de um registro para o outro)
- Ex. mover status de Projeto para ExecucaoProjeto



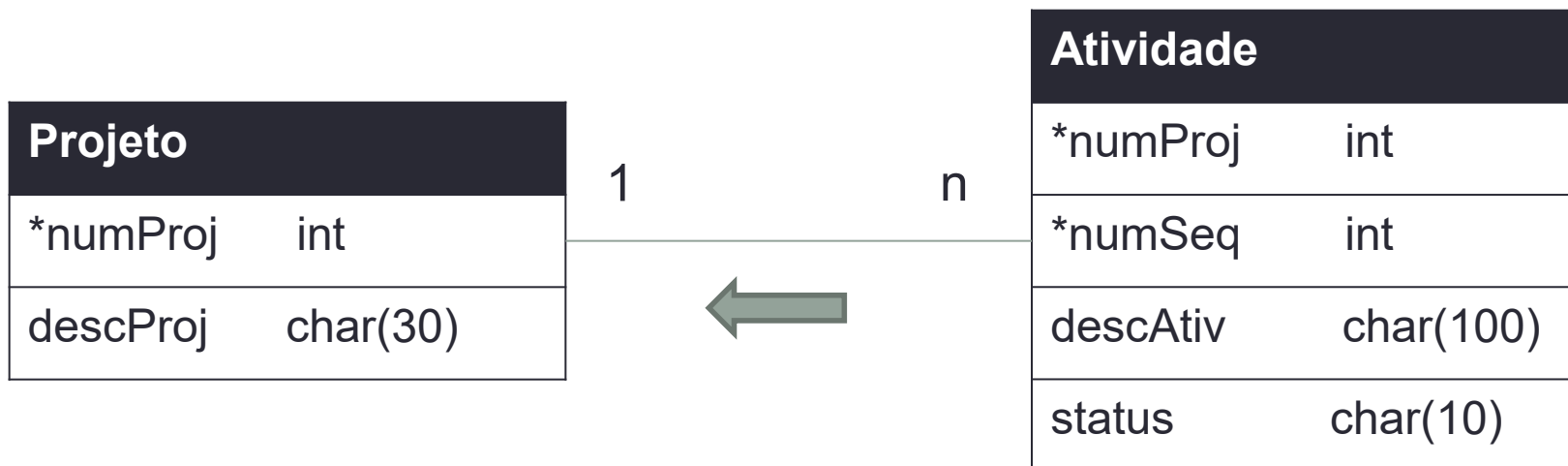
Deslocamento de Coluna

- Quando a cardinalidade é 1-n
 - O script é simples (como vimos anteriormente)
 - Porém, pode gerar dados equivocados
- Ex. mover status de Projeto para Atividade



Deslocamento de Coluna

- Quando a cardinalidade é n-1
 - O script é um pouco mais rebuscado
 - Pode inclusive recorrer a estruturas condicionais da SQL (if, else, ...)
- Ex. mover status de Atividade para Projeto
- o projeto deve ser considerado fechado somente se todas suas atividades estiverem fechadas.



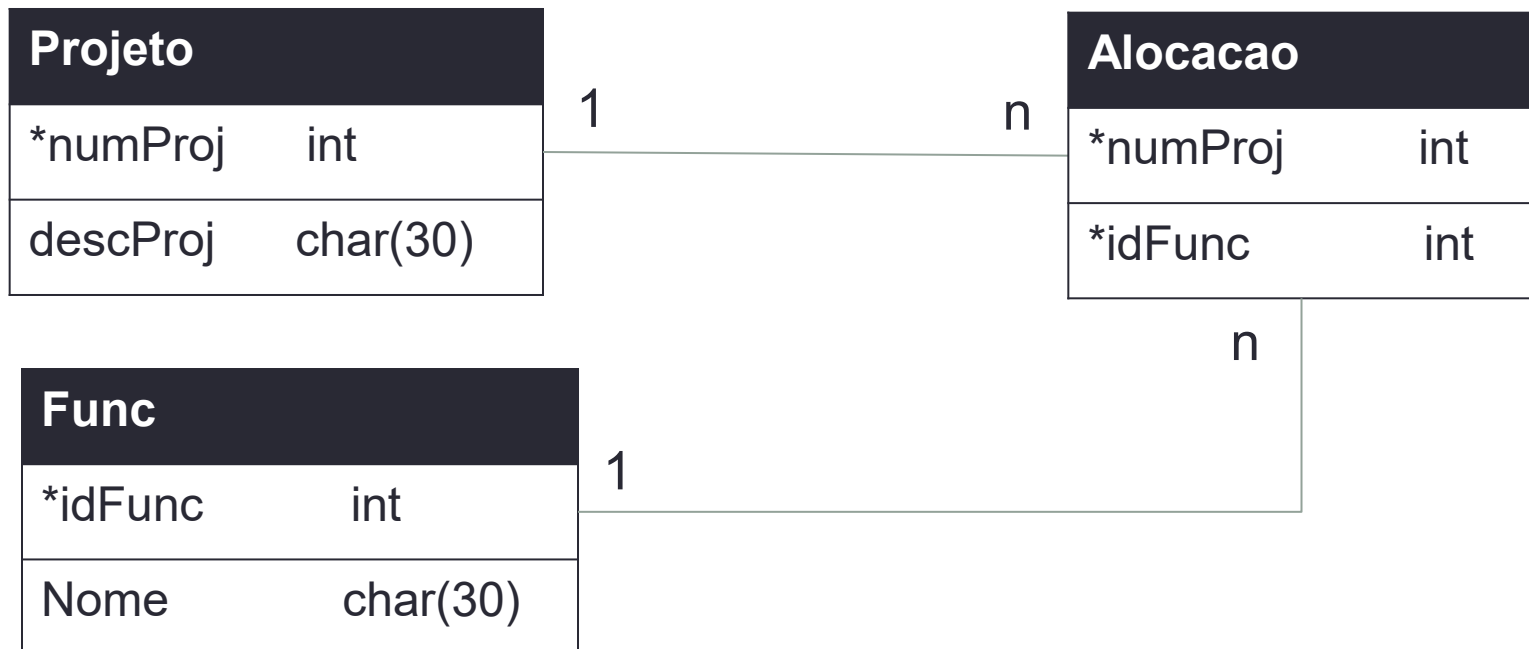
REFATORANDO COLUNAS COM RESTRICÇÕES DE INTEGRIDADE REFERENCIAL

Colunas com Restrições de Integridade Referencial

- Mudanças em colunas são muito mais complexas quando elas possuem alguma restrição de integridade referencial
- Se a coluna pertencer a uma **chave primária**
 - A restrição de chave estrangeira precisa ser removida e depois recriada
- Se além disso a tabela for alvo de alguma **chave estrangeira**
 - o processo é ainda mais complexo
 - Os ajustes devem ser propagados para todas as tabelas que referenciam a tabela alterada
- Tivemos uma noção dessa complexidade quando tratamos com surrogate Keys
 - O exercício a seguir é mais um exemplo que ilustra essa dificuldade

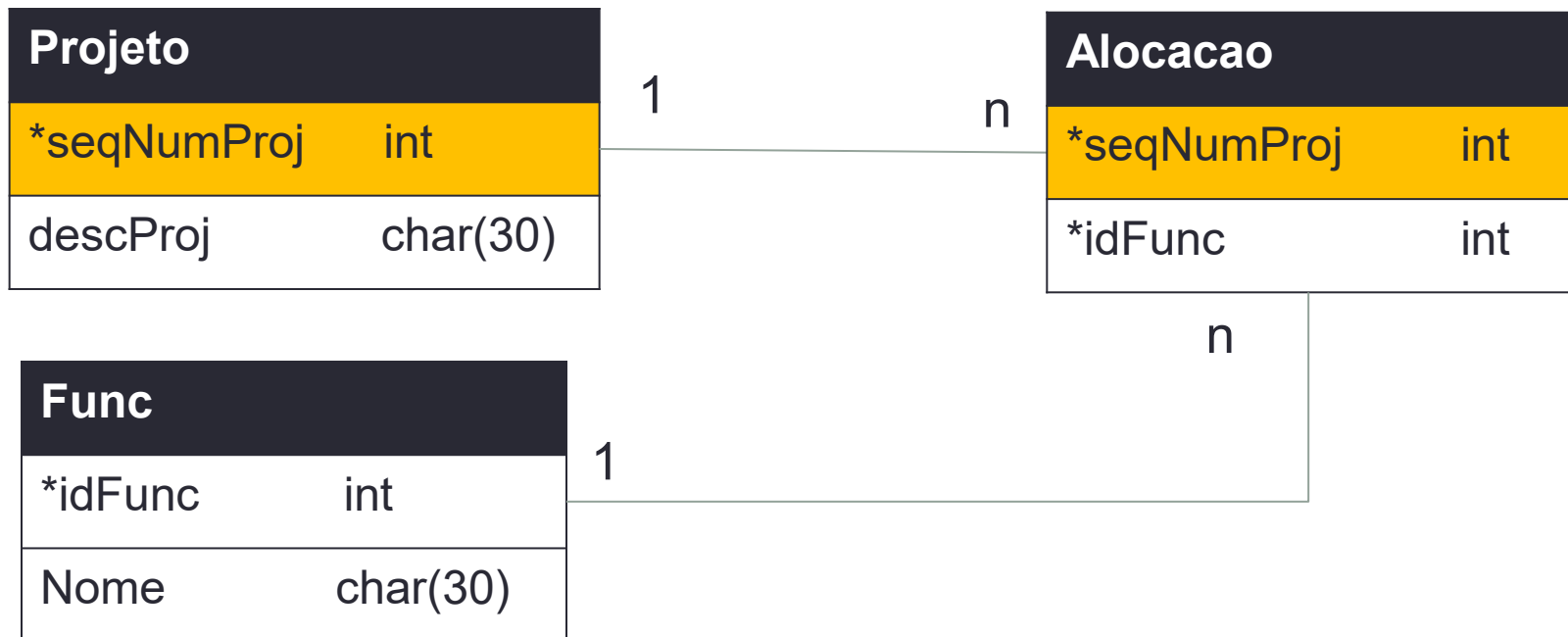
Colunas com Restrições de Integridade Referencial

- O número de projeto é um campo auto incrementável.
- Decidiu-se seguir uma convenção para nomear esse tipo de coluna, prefixando o nome com “seq”



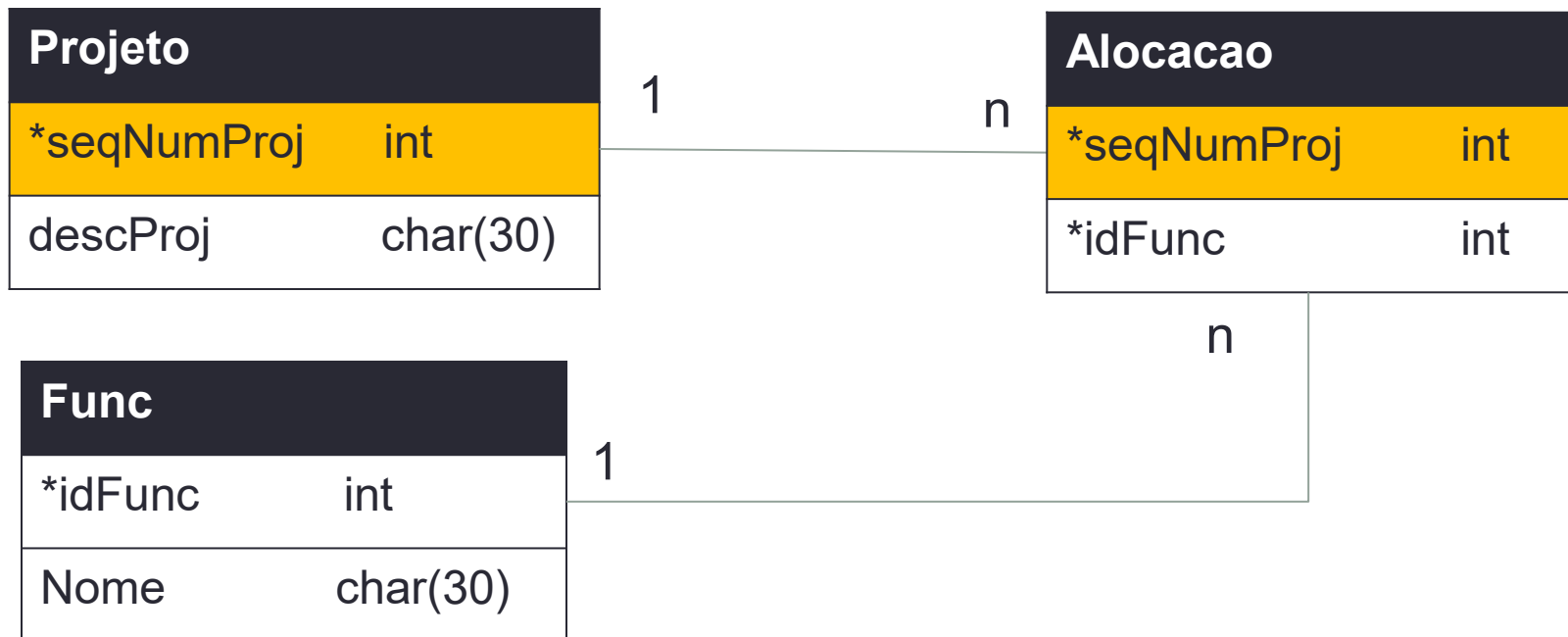
Colunas com Restrições de Integridade Referencial

- Assim deve ficar o modelo



Colunas com Restrições de Integridade Referencial

- Como você implementaria essas modificações sem usar recursos de renomeação de coluna?



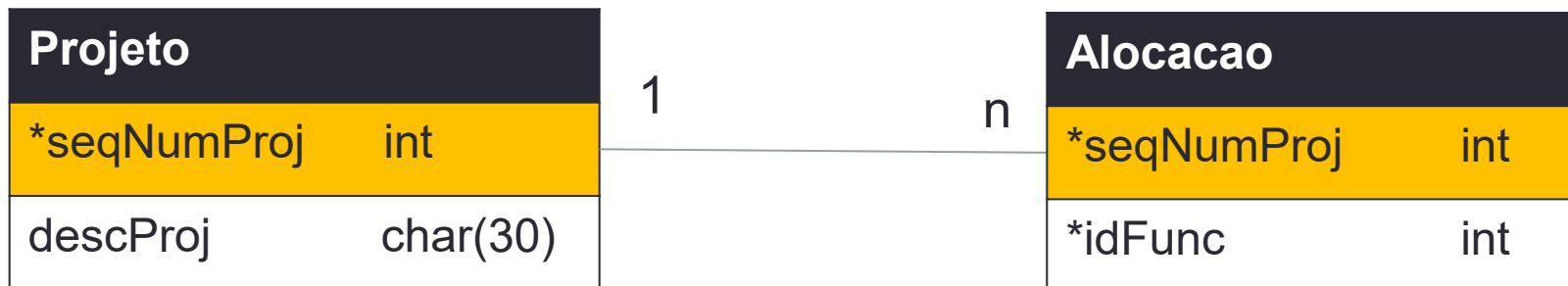
Colunas com Restrições de Integridade Referencial

- Passos (Projeto)

- remover restrição de pk em projeto
- Criar coluna seq em projeto
- Alimentar coluna seq
- Remover coluna velha em projeto
- Criar restrição de pk em projeto

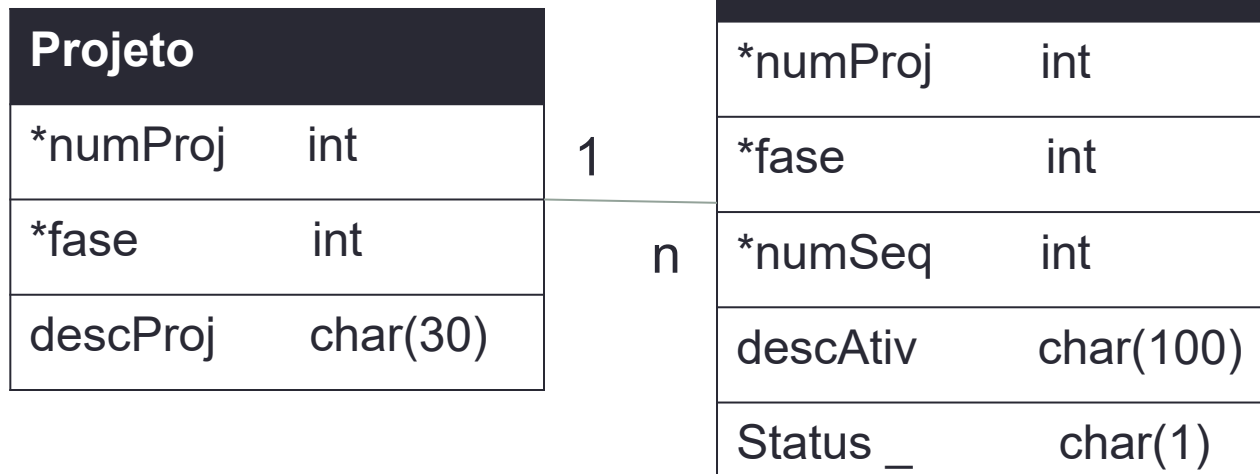
- Passos (Alocacao)

- remover restrição de fk em alocacao
- remover restrição de pk em alocacao
- Criar coluna seq em alocacao
- Alimentar coluna seq
- Remover coluna velha em alocacao
- Criar restrição de pk em alocação
- Criar restrição de fk em alocacao



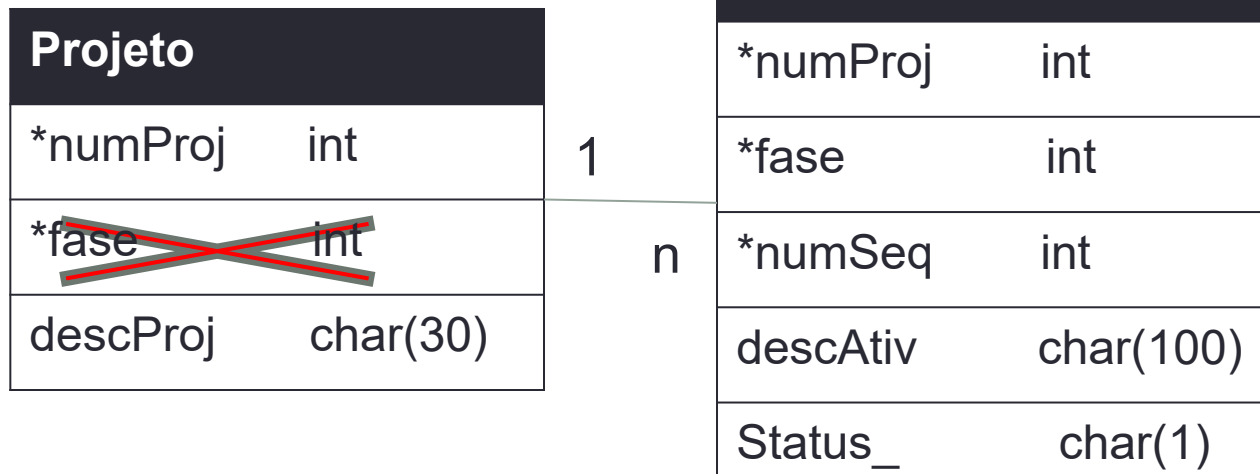
Exercício – Parte 1

- Um projeto possui várias fases. Em cada fase são executadas várias atividades.
- O modelo cria uma instância de “Projeto” para cada fase do projeto, repetindo desnecessariamente a coluna descProj
 - Modelo mal projetado



Exercício – Parte 1

- Um projeto possui varias fases. Em cada fase são executadas várias atividades.
- O modelo cria uma instância de “Projeto” para cada fase do projeto, repetindo desnecessariamente a coluna descProj
 - Modelo mal projetado
- Para resolver isso, decidiu-se eliminar a coluna fase da tabela projeto



Exercício – Parte 2

- Decidiu-se criar uma coluna redundante “status_” em projeto.
- Ele servirá para recuperar mais rapidamente o status de cada projeto
 - O status deve ser encerrado ('E') caso todas atividades tenha status encerrado ('E')
 - O status deve ser aberto ('A') caso pelo menos uma das atividades tenha status aberto ('A')

Projeto	
*numProj	int
*fase	int
descProj	char(30)
Status _	char(1)

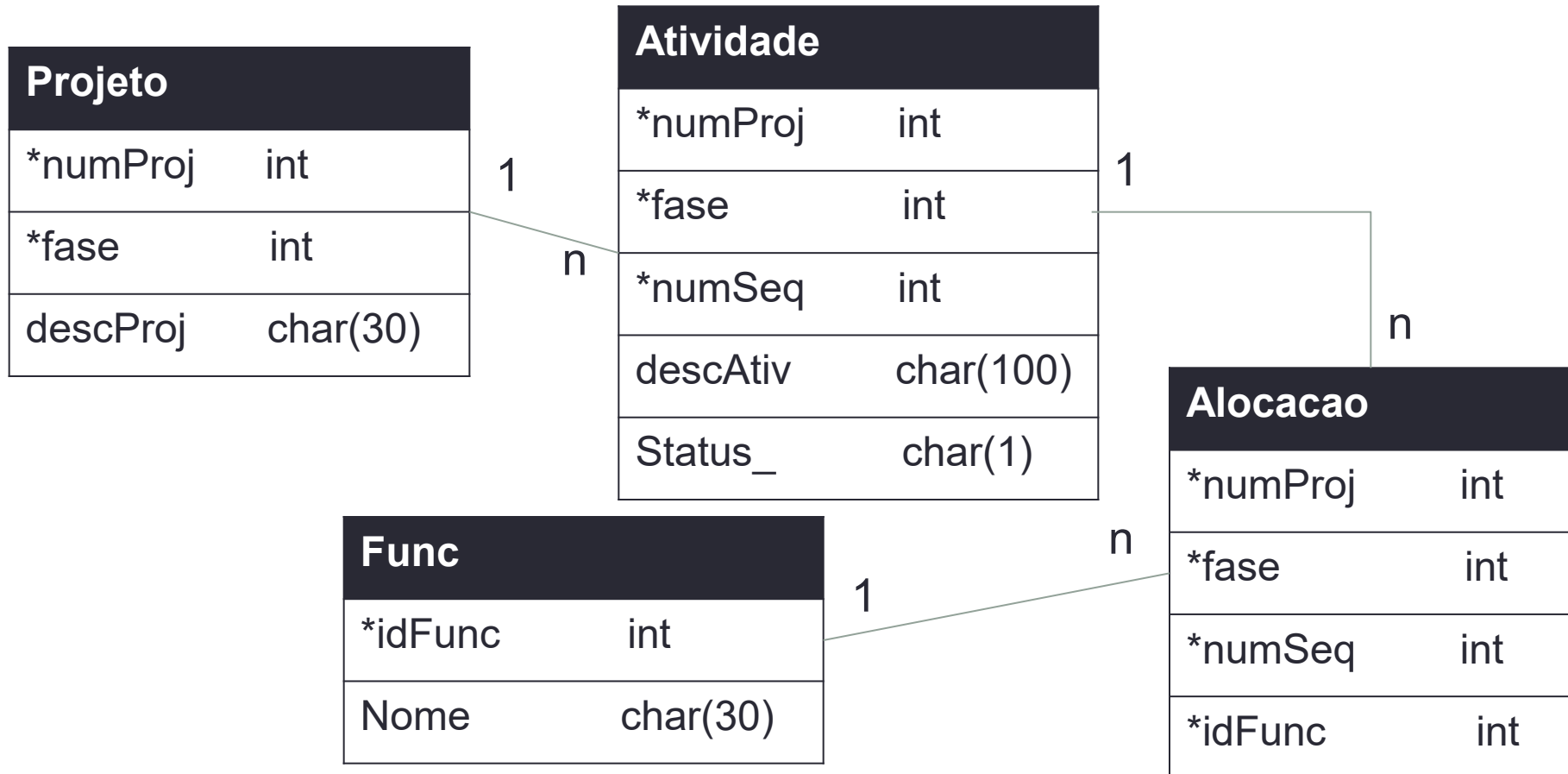
1

n

Atividade	
*numProj	int
*fase	int
*numSeq	int
descAtiv	char(100)
Status_	char(1)

Exercício

- O modelo original completo está descrito abaixo



Exercício

- Como deve ficar

