

REFATORAÇÃO EM BANCOS DE DADOS

Sérgio Mergen

Processos de Refatoração

- Nessa aula veremos os seguintes processos
 - Fusão de Tabela
 - Divisão de Tabela
 - Deslocamento de Dados

Fusão de Tabela

- Os dados de Projeto e Execução do Projeto são mantidos em tabelas separadas
- No entanto, eles costumam ser acessados juntos



Fusão de Tabela

- Os dados de Projeto e Execução do Projeto são mantidos em tabelas separadas
- No entanto, eles costumam ser acessados juntos
- Dessa forma, decidiu-se por fundir essas duas tabelas em uma só

Projeto	
*numProj	int
descProj	char(30)
Status	char(10)
dataIni	date

Fusão de Tabela

- Script para fusão de tabela
 - Passo 1: escolha da tabela a ser mantida
 - Passo 2: deslocamento de colunas para a tabela a ser mantida
 - Passo 3: remoção da tabela obsoleta

Fusão de Tabela

- Script para fusão de tabela
 - Passo 1: escolha da tabela a ser mantida
- Dentre as tabelas a serem fundidas, deve-se escolher a principal para ser mantida
- Ex. Projeto é a tabela principal
 - Como descobrir qual é a principal?

Projeto	
*numProj	int
descProj	char(30)
Status	char(10)

1,1

0,1

ExecucaoProjeto	
*numProj	int
dataIni	int

Fusão de Tabela

- Script para fusão de tabela
 - Passo 1: escolha da tabela a ser mantida
- A tabela principal geralmente é aquela cuja cardinalidade mínima no relacionamento seja 1
 - Essa informação pode ser obtida analisando a documentação(diagramas) ou os registros
- Ex.



Fusão de Tabela

- Script para fusão de tabela
 - Passo 1: escolha da tabela a ser mantida
- Também pode-se descobrir qual é a tabela principal pela posição da chave estrangeira (a tabela principal costuma ser referenciada pelas demais)
- Ex.



Fusão de Tabela

- Script para fusão de tabela
 - Passo 2: deslocamento de colunas para a tabela a ser mantida
- O processo de refatoração “Deslocamento de colunas” pode ser usado para completar esse passo
- Em suma:
 - Criar coluna na tabela destino
 - Migrar valores para a tabela destino
 - Remover coluna na tabela origem

Fusão de Tabela

- Script para fusão de tabela
 - Passo 3: remoção da tabela obsoleta
- Ex.
- `DROP TABLE execucaoProjeto;`

Fusão de Tabela

- A fusão normalmente se aplica quando as tabelas envolvidas possuem relacionamento 1-1
- Desejar fazer isso em tabelas 1-n implica em querer consertar um modelo mal projetado
- Ex.
 - Supostamente um projeto está associado a várias execuções
 - No entanto, na prática um projeto possui apenas uma execução



Processos de Refatoração

- Nessa aula veremos os seguintes processos
 - Fusão de Tabela
 - Divisão de Tabela
 - Deslocamento de Dados

Divisão de Tabela

- Algumas colunas da tabela projeto são raramente acessadas (ex. dataIni)
- Além disso, essas colunas só podem ser acessadas por usuários específicos (ex. gerente de projeto)

Projeto	
*numProj	int
descProj	char(30)
Status	char(10)
dataIni	date

Divisão de Tabela

- Algumas colunas da tabela projeto são raramente acessadas (ex. dataIni)
- Além disso, essas colunas só podem ser acessadas por usuários específicos (ex. gerente de projeto)
- Dessa forma, decidiu-se por dividir essa tabela em duas



- Também conhecido como **particionamento vertical**

Particionamento vertical

- Consiste em separar as colunas de uma tabela em mais do que uma tabela

projeto			
id	desc	status	dataIni
1	Proj1	Aberto	01/2014
2	Proj2	Inativo	04/2014
3	Proj3	Iniciado	03/2015
4	Proj4	Inativo	08/2015

Antes do particionamento

projeto		
id	desc	status
1	Proj1	Aberto
2	Proj2	Inativo
3	Proj3	Iniciado
4	Proj4	Inativo

Depois do particionamento

ExecuçãoProjeto	
id	dataIni
1	01/2014
2	04/2014
3	03/2015
4	08/2015

Divisão de Tabela

- Script para divisão de tabela
 - Passo 1: criação da nova tabela
 - Passo 2: alimentação da nova tabela
 - Passo 3: remoção das colunas obsoletas

Divisão de Tabela

- Script para divisão de tabela
 - Passo 1: criação da nova tabela

- Ex.

```
CREATE TABLE execucaoProjeto (  
    numProj    INT NOT NULL,  
    dataIni    DATE,  
    PRIMARY KEY (numProj),  
    FOREIGN KEY (numProj) REFERENCES projeto (numProj)  
);
```

- A chave primária é a mesma nas duas tabelas
- A chave estrangeira pode ser criada em qualquer uma das duas tabelas
 - Como você decidiria onde incluí-la?

Deslocamento de Dados

- Script para deslocamento de dados
 - Passo 2: alimentação da nova tabela

- Ex.

```
INSERT INTO execucaoProjeto (numProj, dataIni)  
  SELECT numProj, dataIni FROM projeto;
```

Deslocamento de Dados

- Script para deslocamento de dados
 - Passo 2: alimentação da nova tabela
- Ex. (solução alternativa mySQL)

-- criação e alimentação da tabela

```
CREATE TABLE execucaoProjeto AS  
SELECT numProj, dataIni FROM projeto;
```

Obs. O comando acima não recria chaves primárias/estrangeiras
Elas devem ser inseridas manualmente

Divisão de Tabela

- Script para divisão de tabela
 - Passo 3: remoção das colunas obsoletas
- Ex.
`ALTER TABLE projeto DROP dataIni;`

Processos de Refatoração

- Nessa aula veremos os seguintes processos
 - Fusão de Tabela
 - Divisão de Tabela
 - Deslocamento de Dados

Deslocamento de Dados

- A tabela de projetos possui milhões de projetos
 - A maioria deles são projetos que nunca saíram do papel (status = 'Inativo')
- Consultas sobre essa tabela demoram muito para completar
 - No entanto, apenas os projetos executados são de interesse (status <> 'Inativo')

Projeto	
*id	int
Desc	char(30)
Status	char(20)

Deslocamento de Dados

- A tabela de projetos possui milhões de projetos
 - A maioria deles são projetos que nunca saíram do papel (status = 'Inativo')
- Consultas sobre essa tabela demoram muito para completar
 - No entanto, apenas os projetos executados são de interesse (status <> 'Inativo')
- Dessa forma, decidiu-se por deslocar registros de baixa procura para outra tabela

Projeto	
*id	int
Desc	char(30)
Status	char(20)

ProjetoInativo	
*id	int
Desc	char(30)
Status	char(20)

- Processo também conhecido como **Particionamento Horizontal**

Particionamento horizontal

- Consiste em dividir os registros de uma tabela em mais do que uma tabela

projeto		
id	desc	status
1	Proj1	Aberto
3	Proj3	Iniciado
2	Proj2	Inativo
4	Proj4	Inativo

Projeto		
id	desc	status
1	Proj1	Aberto
3	Proj3	iniciado

ProjetoInativo		
id	desc	status
2	Proj2	Inativo
4	Proj4	Inativo

Antes do particionamento

Depois do particionamento

Deslocamento de Dados

- Script para deslocamento de dados
 - Passo 1: criação e alimentação da nova tabela
 - Passo 2: remoção dos registros obsoletos da tabela original

Deslocamento de Dados

- Script para deslocamento de dados
 - Passo 1: criação e alimentação da nova tabela

- Ex.

-- criação da tabela

```
CREATE TABLE projetoInativo (  
    numProj    INT,  
    descProj   char(30),  
    status     char(20),  
    PRIMARY KEY (numProj)  
);
```

-- alimentação da tabela

```
INSERT INTO projetoInativo  
    SELECT * FROM projeto WHERE status = 'Inativo';
```

Deslocamento de Dados

- Script para deslocamento de dados
 - Passo 1: criação e alimentação da nova tabela
- Ex. (solução alternativa MySQL)

-- criação e alimentação da tabela

CREATE TABLE **projetoInativo** AS

SELECT * FROM **projeto** WHERE **status** = 'Inativo';

Deslocamento de Dados

- Script para deslocamento de dados
 - Passo 2: remoção dos registros obsoletos da tabela original
- Ex.

```
DELETE FROM projeto  
WHERE status = 'Inativo';
```

Deslocamento de Dados

- E se os registros de projeto forem referenciados por outras tabelas?
- Ao remover os registros inativos da tabela Projeto
 - Pode-se violar alguma restrição de chave estrangeira

Projeto	
*numProj	int
Status	char(20)

ProjetoInativo	
*numProj	int
Status	char(20)

Atividade	
*numProj	int
*numAtiv	int
Desc	char(200)

1 n

Deslocamento de Dados

- Script para deslocamento de dados referenciados
 - Solução 1
 - Remover registros que possuem a chave estrangeira
- Ex.

```
DELETE FROM atividade WHERE numProj IN  
      (SELECT numProj FROM projeto WHERE status = 'Inativo');
```

- Problema: Leva a perda de dados

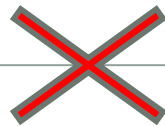
Deslocamento de Dados

- Script para deslocamento de dados referenciados
 - Solução 2
 - Remover restrições de chave estrangeira

- Ex.

Projeto	
*numProj	int
Status	char(20)

1



n

Atividade	
*numProj	int
*numAtiv	int
Desc	char(200)

ProjetoInativo	
*numProj	int
Status	char(20)

Deslocamento de Dados

- Script para deslocamento de dados referenciados
 - Solução 2
 - Remover restrições de chave estrangeira

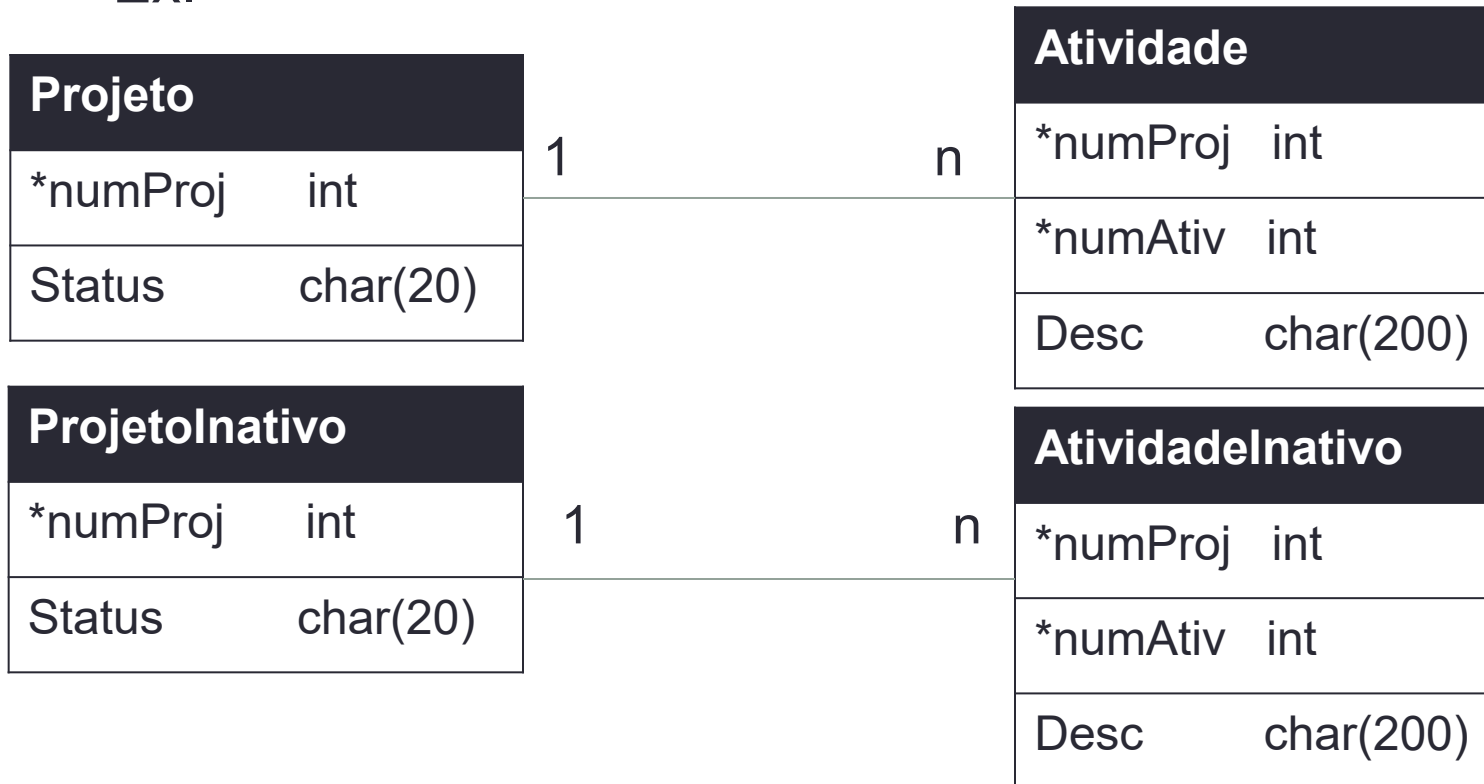
- Ex.

```
ALTER TABLE atividade DROP CONSTRAINT fk_ativ_proj;
```

- Problema:
 - Restrição de integridade referencial precisa ser garantida via aplicação

Deslocamento de Dados

- Script para deslocamento de dados referenciados
 - Solução 3
 - Duplicar a estrutura em volta da tabela particionada
- Ex.



Deslocamento de Dados

- Script para deslocamento de dados referenciados
 - Solução 3
 - Duplicar a estrutura em volta da tabela particionada

- Ex.

```
CREATE TABLE projetoInativo AS SELECT ...  
CREATE TABLE atividadeInativo AS SELECT ...  
ALTER TABLE projetoInativo ADD CONSTRAINT ...  
ALTER TABLE atividadeInativo ADD CONSTRAINT ...  
DELETE FROM atividade WHERE ...  
DELETE FROM projeto WHERE ...
```

- Problema:
 - Modelos maiores podem ser mais difíceis de entender
 - Gera mais de um ponto de mudança caso se precise alterar o esquema

Deslocamento de Dados

- Script para deslocamento de dados referenciados
 - Solução 4
 - Relaxar restrições de chave estrangeira
- Ex.

ProjetoAtivo	
*numProj	int
Status	char(20)

0,1

n

ProjetoInativo	
*numProj	in
Status	char(20)

0,1

Atividade	
*numSeq	int
numAtiv	int
numProjAtivo	int
numProjInativo	int
Desc	char(200)

n

Deslocamento de Dados

- Script para deslocamento de dados referenciados
 - Solução 4
 - Relaxar restrições de chave estrangeira

• Ex.

```
CREATE TABLE atividade (  
    numSeq          INT AUTO_INCREMENT PRIMARY KEY ,  
    numAtiv         INT      NOT NULL,  
    numProjAtivo    INT      NULL,  
    numProjInativo  INT      NULL,  
    desc            char(30) NOT NULL,  
    FOREIGN KEY numProjAtivo    REFERENCES projeto (numProj)  
    FOREIGN KEY numProjInativo  REFERENCES projetoInativo (numProj)  
);
```

- Problema 1: aceita atividades sem vínculo a nenhum projeto
 - Ambas fks aceitam nulos

Deslocamento de Dados

- Script para deslocamento de dados referenciados
 - Solução 4
 - Relaxar restrições de chave estrangeira

• Ex.

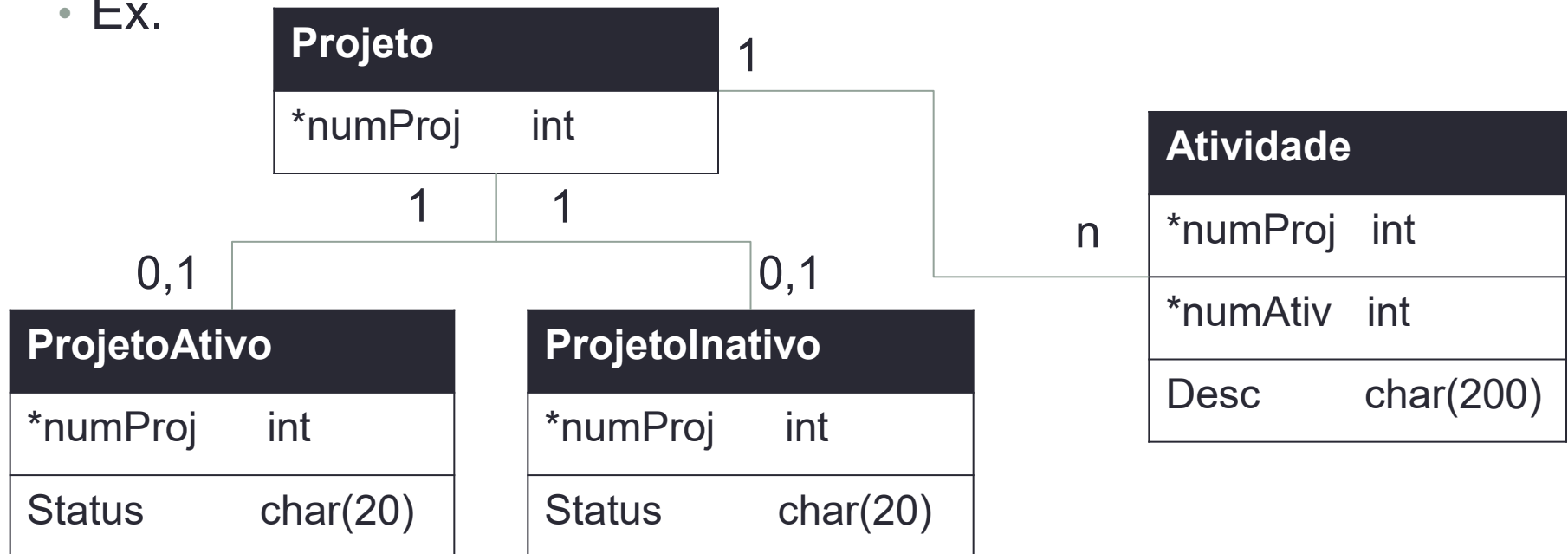
```
CREATE TABLE atividade (  
    numSeq          INT AUTO_INCREMENT PRIMARY KEY ,  
    numAtiv         INT      NOT NULL,  
    numProjAtivo    INT      NULL,  
    numProjInativo  INT      NULL,  
    desc            char(30) NOT NULL,  
    FOREIGN KEY numProjAtivo    REFERENCES projeto (numProj)  
    FOREIGN KEY numProjInativo  REFERENCES projetoInativo (numProj)  
);
```

- Problema 2: O processo de refatoração da tabela “atividade” é custoso
 - Envolve mudança em chave primária

Deslocamento de Dados

- Script para deslocamento de dados referenciados
 - Solução 5
 - Usar tabelas especializadas

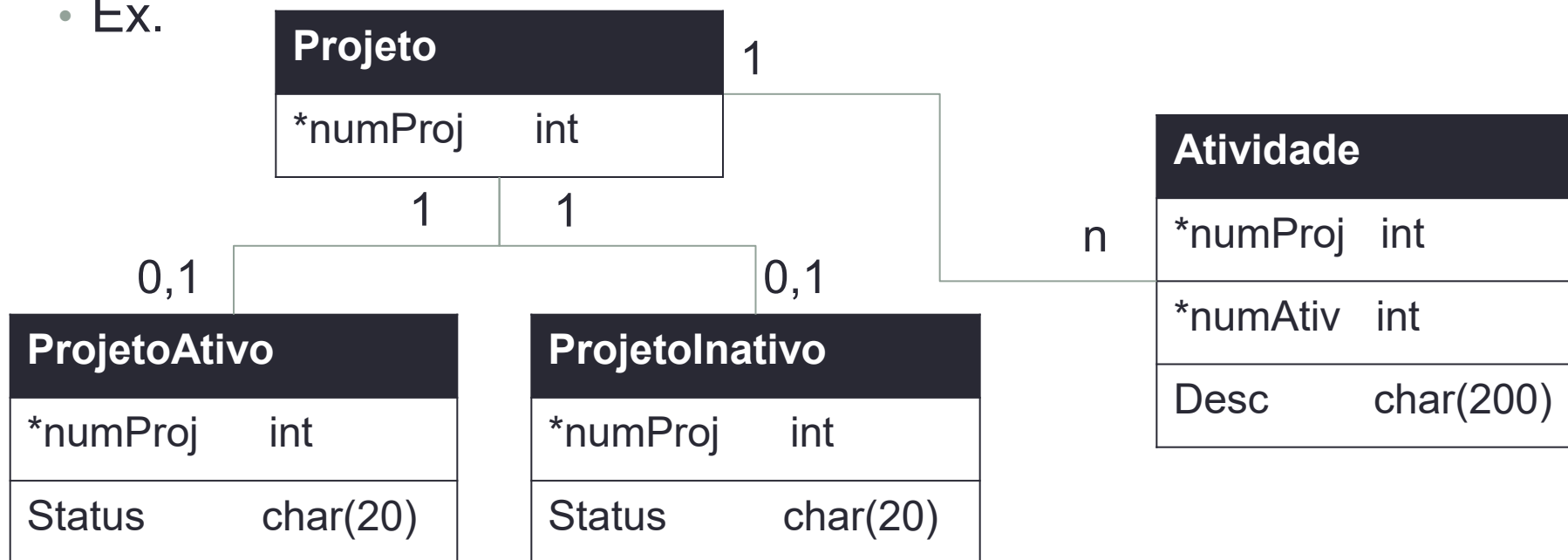
• Ex.



Deslocamento de Dados

- Script para deslocamento de dados referenciados
 - Solução 5
 - Usar tabelas especializadas

• Ex.



- Problema: o mesmo numProj pode ser usado para projetos ativos e inativos
 - Esse controle de integridade deve ser feito de outras maneiras (ex. via triggers)

Deslocamento de Dados

- Script para deslocamento de dados referenciados
 - Solução 6
 - Usar particionamento horizontal via SGBD
- Ex.

ALTER TABLE **projeto**

PARTITION BY LIST COLUMNS (**status**)(

PARTITION **pInativo** VALUES IN ('inativo'),

//todos os demais valores devem aparecer aqui

PARTITION **pAtivo** VALUES IN ('aberto', 'iniciado', 'ativo'),

)

PARTICIONAMENTO HORIZONTAL VIA SGBDS

Particionamento horizontal via SGBDs

- Consiste em dividir os registros de uma tabela em partições
 - Com base em alguma coluna de segregação
 - Ex. status

projeto		
id	desc	status
1	Proj1	Aberto
2	Proj2	Inativo
3	Proj3	Iniciado
4	Proj4	Inativo

Sem partições

Projeto (partição 1)		
id	desc	status
1	Proj1	Aberto
3	Proj3	iniciado

Projeto (partição 2)		
id	desc	status
2	Proj2	Inativo
4	Proj4	Inativo

Com partições

Particionamento horizontal via SGBDs

- No particionamento horizontal manual
 - o DBA gera uma tabela por partição
- no particionamento via SGBD
 - apenas uma tabela é mantida (o DBA enxerga apenas uma tabela)
 - E essa tabela é dividida em partições

projeto		
id	desc	status
1	Proj1	Aberto
2	Proj2	Inativo
3	Proj3	Iniciado
4	Proj4	Inativo

Sem partições

Projeto (partição 1)		
id	desc	status
1	Proj1	Aberto
3	Proj3	iniciado

Projeto (partição 2)		
id	desc	status
2	Proj2	Inativo
4	Proj4	Inativo

Com partições

Particionamento horizontal via SGBDs

- Sintaxe mySQL para criar tabela com partições

`CREATE TABLE tabela (...) ENGINE=INNODB PARTITION BY ...;`

- Também pode-se criar partições em uma tabela já existente

`ALTER TABLE tabela PARTITION BY ...;`

Particionamento horizontal via SGBDs

- Alguns tipos de particionamento possíveis em MySQL
 - PARTITION BY LIST COLUMNS
 - PARTITION BY KEY
 - PARTITION BY RANGE
- **Obs.** Só funciona se a coluna particionada for parte da chave primária

Particionamento horizontal via SGBDs

- Exemplo usando **PARTITION BY LIST COLUMNS**

- Atividade (estado, idProj, desc)

-- Dividindo em 2 partições, separando as atividades de acordo com o estado onde elas foram executadas

PARTITION BY LIST COLUMNS (estado)(

PARTITION pinterno VALUES IN ('rs'),

-- todos os demais valores devem aparecer aqui

PARTITION pExterno VALUES IN ('sc', 'sp', 'rj'),

)

Particionamento horizontal via SGBDs

- Exemplo usando **PARTITION BY KEY**
- Atividade (dataIni, idProj, desc)
 - Dividindo registros em 6 partições, pela data inicial da atividade
PARTITION BY KEY (dataIni) PARTITIONS 6;

Particionamento horizontal via SGBDs

- Exemplo usando **PARTITION BY RANGE**

- Atividade (dataIni, idProj, desc)

-- Dividindo registros em 10 partições, pelo década de início da atividade

PARTITION BY RANGE(YEAR (**dataIni**)) (

PARTITION **p1** VALUES LESS THAN (1960),

PARTITION **p2** VALUES LESS THAN (1970),

...

PARTITION **p10** VALUES LESS THAN MAXVALUE,

);

Deslocamento de Dados

- Na verdade, a solução por particionamento horizontal não funciona no caso do exemplo
 - Projeto(id, desc, status)
 - Por quê?

ALTER TABLE **projeto**

PARTITION BY LIST COLUMNS (**status**)(

PARTITION **pInativo** VALUES IN ('inativo'),

//todos os demais valores devem aparecer aqui

PARTITION **pAtivo** VALUES IN ('aberto', 'iniciado', 'ativo'),

)

Particionamento horizontal via SGBDs

- **Vantagens**

- Dados obsoletos podem ser descartados ou arquivados facilmente.
- Consultas que filtram pela coluna de partição são mais eficientes.
- Se as partições estiverem em discos diferentes
 - **Maior vazão**
 - Possibilidade de **paralelizar consultas**

Particionamento horizontal via SGBDs

- **Desvantagens**

- Nem toda coluna é adequada para particionamento.
- Se todas as partições estiverem no **mesmo disco**, o ganho de performance é limitado.
- Consultas que acessam muitas partições podem ficar mais lentas.
- Atualizações exigem descobrir **em qual partição o registro deve ficar**.

Exercício 1

- Qual tabela seria melhor particionar?
- Qual coluna?
- Que função de particionamento usar?

Func (idFunc, nome, salario, contratacao, demissão)

Proj (idProj, nome, dataInicio, dataTermino)

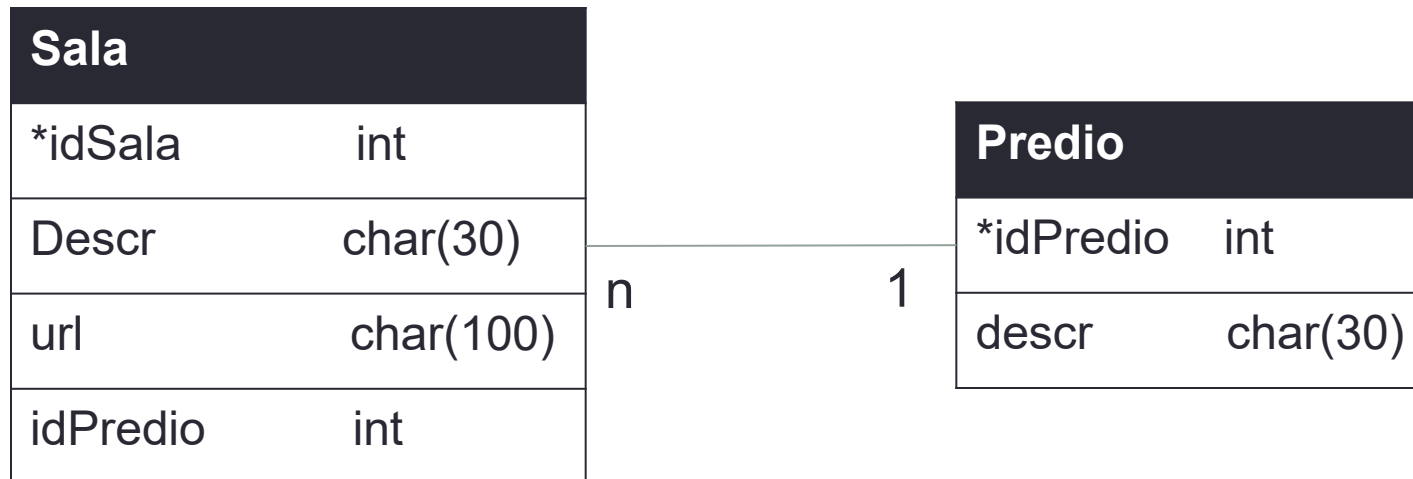
Aloc (idAloc, idFunc, idProj, funcao)

Atividade(data,idAloc,descricao)

- Obs.
 - Existem 5.000 funcionários, e 50.000 projetos.
 - Cada projeto possui em média 5 funcionários
 - Em cada projeto um funcionário realiza em média 100 tarefas

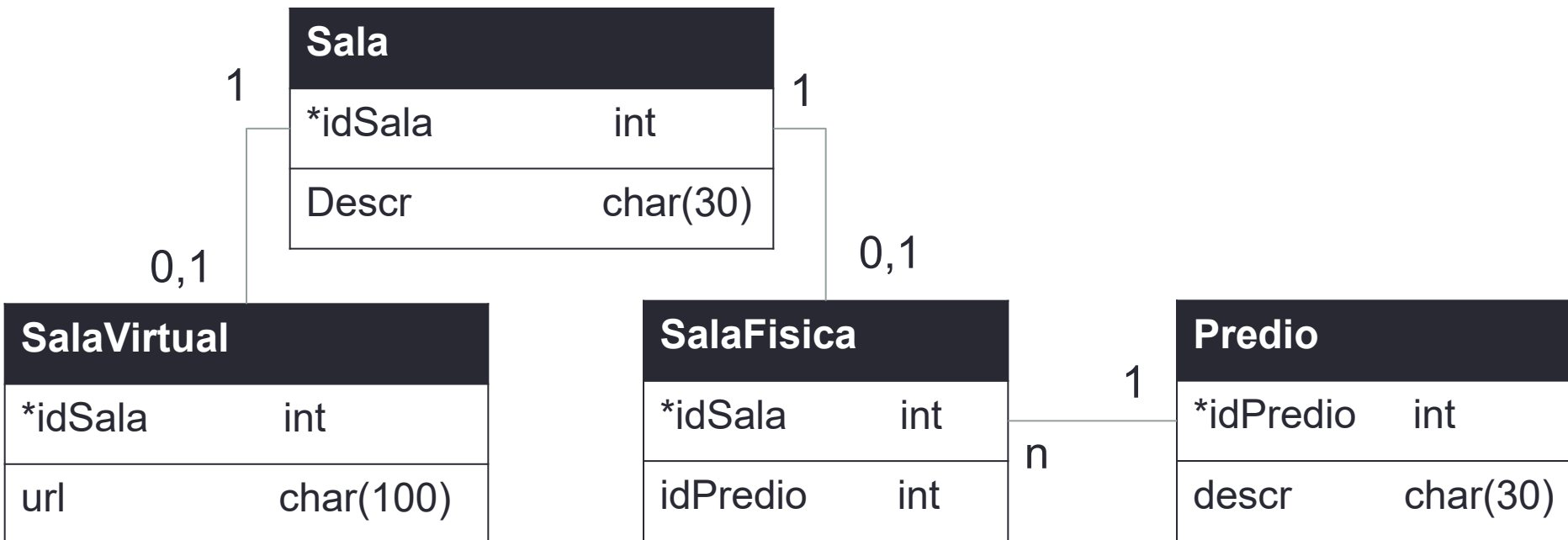
Exercício 2

- Uma sala pode ser virtual ou física.
- A url é importante apenas para salas virtuais
- Apenas salas físicas têm um prédio associado.



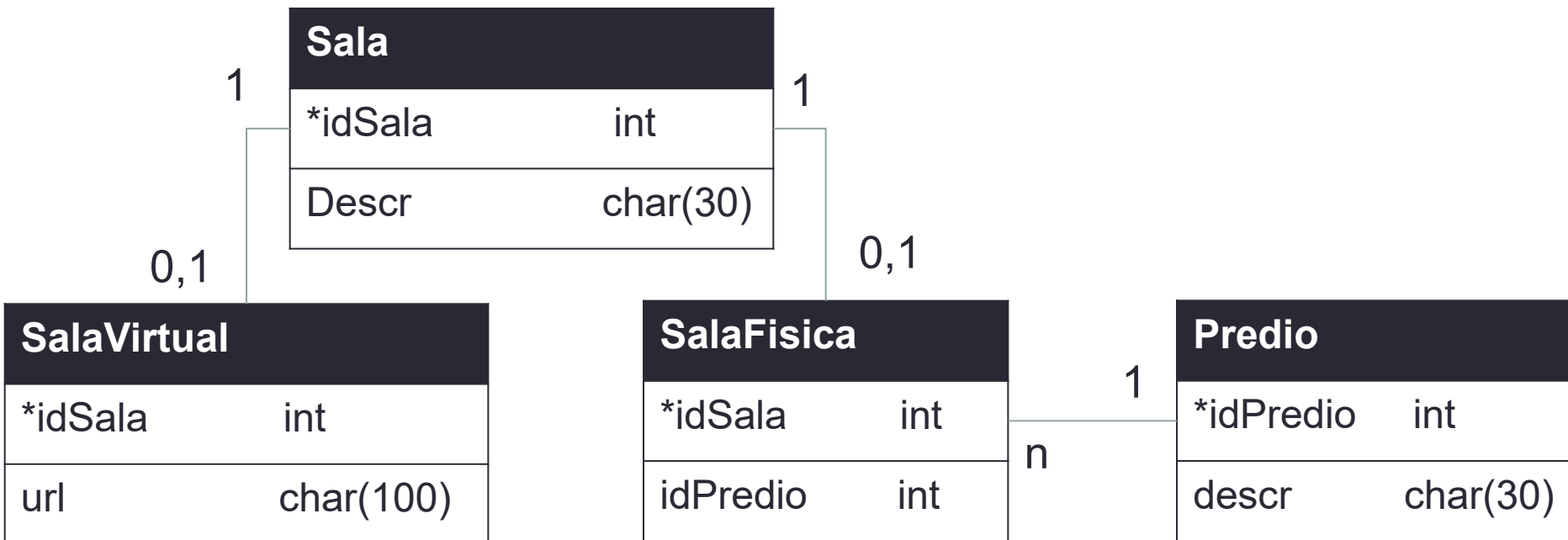
Exercício 2

- Uma sala pode ser virtual ou física.
- A url é importante apenas para salas virtuais
- Apenas salas físicas tem um prédio associado.
- Dessa forma, decidiu-se passar a usar uma hierarquia de salas.



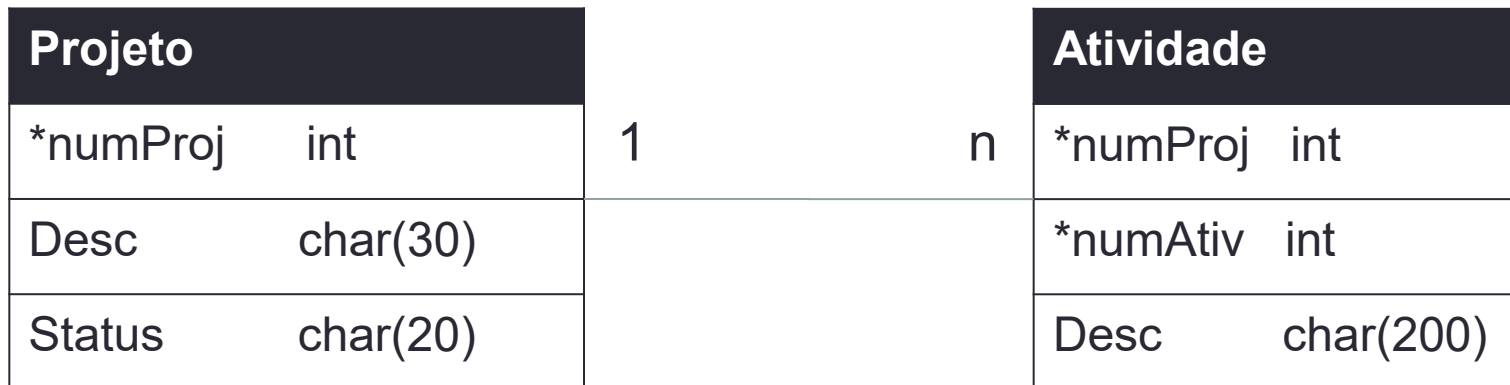
Exercício 2

- Como você resolveria isso?



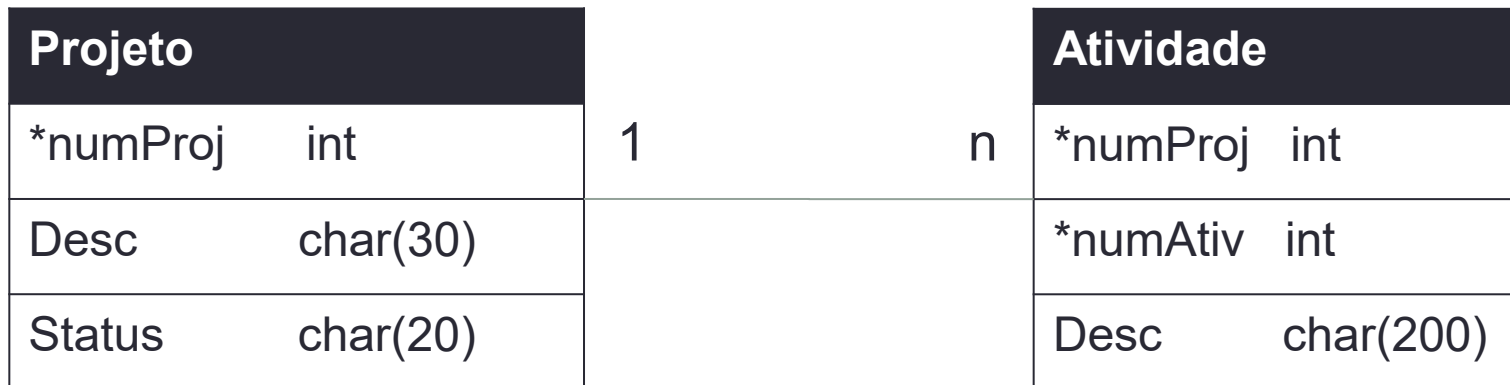
Atividade Individual

- Há dois tipos de projeto: ativos e inativos
- Decidiu-se aplicar o processo de particionamento horizontal para retirar os projetos inativos da tabela de projeto, movendo-os para a tabela **ProjetoInativo**



Atividade Individual

- No entanto, isso gera um problema
 - a tabela atividade faz referência à tabela projeto
 - se movermos os projetos para outra tabela, perderemos a integridade referencial.



Atividade Individual

- Solução a ser adotada: usar tabelas especializadas
 - A tabela atividade manterá uma chave estrangeira referenciando a tabela genérica projeto
 - preservando assim a integridade referencial.

