

# MAC0242 - Laboratório de Programação II

prof. Dr. Alfredo Goldman

*Terceiro mini exercício programa*

## Segunda etapa

Escolha uma das duas opções:

### 1. Precedência de restauração de tabelas em bancos de dados

Seu primeiro dia como sysadmin pythonzeiro e, de repente, o banco de dados cai! Pior: os registros foram corrompidos.

O sysadmin anterior deixou o backup como um *dump* de todas as tabelas do banco de dados em questão. Ou seja, há um arquivo contendo a estrutura de todas as tabelas chamado "estrutura.sql" e diversos arquivos nomeados como <nome da tabela>.sql contendo os dados brutos.

O problema a se resolver é determinar a ordem para se repovoar os dados, pois há uma possível relação de dependência de dados entre o conteúdo das tabelas. Por exemplo, se tivéssemos uma tabela contendo o clã de jogadores de Tibia, eventualmente precisaríamos repovoar primeiro a tabela de jogadores para finalmente poder repovoar a tabela de clãs.

No arquivo de estrutura, uma dependência é descrita pela expressão:

```
CONSTRAINT `<nome da restrição>` FOREIGN KEY  
(`<campo>`) REFERENCES `<nome da tabela de origem>`  
(`<campo da tabela de origem>`)
```

#### **Entrada:**

Arquivo com a estrutura do banco de dados passado por parâmetro.

#### **Saída:**

- Uma lista com os nomes de arquivos <nome da tabela>.sql na ordem que poderão repovoar o banco.

### Exemplo de execução:

```
root@mysql:/# ls -l /tmp/trabalho/urgh
estrutura.sql
tabela1.sql
tabela3.sql
tabela4.sql

root@mysql:/# script /tmp/trabalho/urgh
tabela3.sql
tabela1.sql
tabela4.sql
```

**Observação:** quem tiver interesse em saber como se faz dump de banco de dados em questão, particularmente o do exemplo, segue o link:

<http://dev.mysql.com/doc/refman/5.1/en/mysqldump.html>

## 2. Revertendo DNS

Domain Name System (DNS), como o nome diz, é um sistema responsável por converter endereços IP em nomes. Quando tentamos acessar "www.youtube.com", um servidor DNS converte "www.youtube.com" num endereço IP e, depois, consegue acessar.

Quando estamos administrando esses servidores de DNS, há um problema bastante chato e razoavelmente propenso a erros: precisamos manter, também, os registros de DNS reverso. Como o próprio nome diz, DNS reverso é o contrário do DNS: ele recebe um IP e devolve um nome.

Logo, num servidor DNS, teremos dois arquivos - doravante denominados arquivos de zona:

- DNS "direto";
- DNS reverso.

É preciso, pois, escrever um script que processe um arquivo de zona e devolva o reverso a partir dele. Isto é:

- copiar o cabeçalho;
- copiar as entradas do tipo NS;
- e, para cada entrada do tipo A, deveremos ter uma entrada do tipo PTR.

#### Formato de entrada do DNS:

```
<nome do host>      A      <IP>
```

#### Formato de entrada do DNS reverso:

```
<última parte do IP>  PTR  <nome completo>
```

Um arquivo de zona contém um cabeçalho como o abaixo:

```
$TTL 604800
@      IN      SOA      ns.dominio.exemplo. admin.dominio.exemplo (
                                2014090601 ; Serial
                                604800    ; Refresh
                                86400     ; Retry
                                2419200   ; Expire
                                604800 )   ; Negative Cache TTL
```

Note que o serial costumeiramente é baseado na data e num número inteiro. Entretanto, para fazer este exercício, essa informação é irrelevante. O relevante é que abaixo do cabeçalho teremos as entradas como a seguir:

```
      NS      ns.dominio.exemplo.
gateway  A      192.168.240.2
mail     A      192.168.240.13
ns       A      192.168.240.4
shell    A      192.168.240.3
www      A      192.168.240.9
```

É possível também termos entradas como MX, CNAME, SRV, TXT etc. Mas não trabalharemos com elas porque elas são *desnecessárias* para o reverso.

### Exemplo de entradas do DNS reverso:

```
NS    ns.dominio.exemplo.  
2     PTR    gateway.dominio.exemplo.  
3     PTR    shell.dominio.exemplo.  
4     PTR    ns.dominio.exemplo.  
9     PTR    www.dominio.exemplo.  
13    PTR    mail.dominio.exemplo.
```

## Testes

Vocês também deverão entregar um script shell (ou python) que valide automaticamente a saída do mini-EP, para algumas entradas. Nesse script, vocês devem fornecer uma entrada, rodar o programa e avaliar a saída. Se a saída for a esperada, o script deve imprimir na saída padrão “O teste passou com sucesso!”. Caso contrário, ele deverá imprimir “O teste falhou! =(“.

Procure sempre testar casos extremos como a entrada vazia ou uma entrada muito grande. Geralmente programas quebram com elas. Bem como, teste o caso médio para garantir.

Não se esqueça de junto com o script enviar todas as entradas necessárias para os testes.

## Relatório

Junto com o programa escolhido, é esperado um relatório no arquivo README contendo os nomes e número USP da dupla (ou pessoa), especificando qual foi a opção e demais informações que julgarem importantes para a correção. Pode ser um simples arquivo texto.

## Importante

- O grupo deverá, inclusive, pesquisar módulos/bibliotecas que possam ajudá-los a resolver os exercícios como, por exemplo, uma biblioteca do Python que

processe requisições HTTP, outra que processe argumentos por linha de comando ou mesmo uma que trabalhe com expressões regulares.

- O trabalho é estritamente em dupla ou individual. Veja a [política do Departamento de Ciência da Computação para casos de plágio ou cola](#).
- Escreva de forma clara e estruturada todos os seus códigos e relatórios. Organize e nomeie todos os arquivos entregues de forma que eles possam ser facilmente identificados. A avaliação levará em conta todas essas questões! Uma apresentação ruim, ou a falta de clareza, poderá prejudicar sua nota.
- O programa deve ser entregue por meio do sistema [Paca](#).
- Todos os arquivos devem ser entregues em um arquivo zip no formato **mini-ep3-<seu-número-USP>-<número-USP-dupla>.zip**. Exemplo: Se seu número USP for 12345678, você deverá entregar um arquivo com o nome mini-ep3-12345678.zip. **Apenas uma pessoa deve fazer essa entrega.**
- Enquanto o prazo de entrega não expirar, você poderá entregar várias versões do mesmo mini exercício-programa. Apenas a última versão entregue será guardada pelo sistema. Encerrado o prazo, você perderá 1 ponto para cada hora de atraso. Não deixe para entregar seu exercício na última hora!
- Guarde uma cópia do seu mini exercício-programa pelo menos até o final do semestre.