

## ARTIGO: 11488

### Importação de dados

A importação de dados é uma etapa que faz parte do processo de ETL (Extract, Transform and Load) de um BI.

Essa etapa de extração foi desenvolvida em Python pela flexibilidade e facilidade da linguagem, já que esse processo consiste apenas em extrair os dados do banco de dados, sem a necessidade de muitos tratamentos, salvar em um arquivo e enviar para o servidor, onde os dados serão transformados e carregados para o uso do BI.

Os arquivos para importação são encontrados em: <http://bi.hos.com.br/files/import.zip>

Baixe também o conector firebird para python em <http://bi.hos.com.br/files/pyfirebirdsql.zip>

### Instalando e configurando o importador

1 - Instale o Python versão 2.7 <http://bi.hos.com.br/files/python-2.7.10.amd64.msi>

2 - Instale o conector firebird. Abra o CMD e navegue até a pasta do pyfirebirdsql e execute

```
python.exe setup.py install
```

Nesse zip os seguintes arquivos podem ser encontrados:

**config.py** -> Nesse arquivo são configurados os caminhos do banco de dados de onde os dados serão extraídos, as pastas de destino para os dados extraídos e as informações de acesso aos BI WEB

```
server_info = {
    'username': 'usuario_etl@farmacia.com.br', # usuario com acesso ao BI no grupo ETL
    'password': '@123', # senha de acesso ao BI
    'base_url': 'http://bi.hos.com.br/data/', # url base do BI + /data/
    'incremental': False, # se importação é nova ou incremental
}

database_info = {
    'host': 'localhost', # host do banco de dados
    'port': '3050', # porta do banco de dados
    'path': 'c:\merchfarma\cadastro.gdb', # caminho do banco para extração dos dados
    'username': 'SYSDBA', # usuario do banco
    'password': 'masterkey', # senha do banco
    'charset': 'ISO8859_1', # charset do banco
}

export_info = {
    'path_sql_file_to_import': 'c:\bi\data_import.sql', # arquivo com sql's modelo para a extração dos dados
    'path_data_export': 'c:\bi\data' # pasta para salvar os arquivos gerados
}
```

**db\_extract.py** e **fbdatabase.py** -> Scripts que realizam a importação

### Iniciar a importação

Para iniciar a importação execute em um terminal:

```
python db_extract.py
```

Você poderá acompanhar o status da importação no terminal. Para a primeira importação, configure a variável **incremental = False**, e para as próximas importações configure **incremental = True**.

Após o envio dos dados o servidor web saberá que deve processar os dados recebidos. **Após esse processamento realizado o usuário ETL receberá um email com o status e logs do processamento da importação.**

### Definição de importação

A definição do que será importado fica no arquivo **data\_import.sql** e deve seguir a seguinte estrutura:

```
output clientes.data
select codigo ||'|'|| nome ||'|'||
case when data_nasc is null then '1900-01-01' else data_nasc end
```

```
||'|'||
case upper(sexo)
  when 'M' then 'Masculino'
  when 'F' then 'Feminino'
  else 'Não Informado'
end
from clientes;
```

Antes de cada sql temos que informar o nome do arquivo de destino para os dados. Nesse caso, os clientes serão armazenados no arquivo **clientes.data** no diretório **c:\bi\data**. Os registros devem obrigatoriamente usar o separador | (pipe). Uma parte do tratamento dos dados pode ser feita nessa extração inicial.

Devemos olhar para os arquivos:

**db\_create\_metadata.sql** -> Estrutura do banco de dados do BI

**after.sql** -> SQL's executadas após a importação no banco do BI

**before.sql** -> SQL's executadas antes da importação no banco do BI

Esses arquivos são configurados na criação da empresa no BI.

Os dados extraídos do banco de dados devem estar de acordo com o arquivo de criação de banco de dados usado pelo BI, que pode ser encontrado no cadastro da empresa (**db\_create\_metadata.sql**) e também com os scripts de importação (apache pig).

No caso de importação incremental deve ser adicionada a cláusula de data para o filtro. Todos os registros podem ser incrementais ou apenas uma parte deles. Toda a tabela que for incremental deve ser assim "marcada" no arquivo **before.sql**:

Arquivo **before.sql**

```
delete from dimensao_produto
delete from fato_vendas,incremental
```

Aqui temos para exemplo uma parte do arquivo, onde a tabela **dimensao\_produto** não é incremental, isso é, ela será completamente removida antes da importação. Já a **tabela fato\_vendas** é incremental, ou seja, os dados serão parcialmente removidos de acordo com o valor configurado em '**Dias para atualização incremental**' nas configurações da empresa no BI.

Então, se quisermos uma importação incremental de 30 dias, temos que informar isso nas configurações da empresa no BI (configuração 'Dias para atualização incremental') e também configurarmos isso no arquivo **data\_import.sql** de modo que os dados extraídos respeitem esse filtro de data (ex.: *where data >= data - 30*).