TDW BI Consulting

Testes Unitários

O fim do começo

Escrito por: Mikael Ducatti Walczak

08/2025

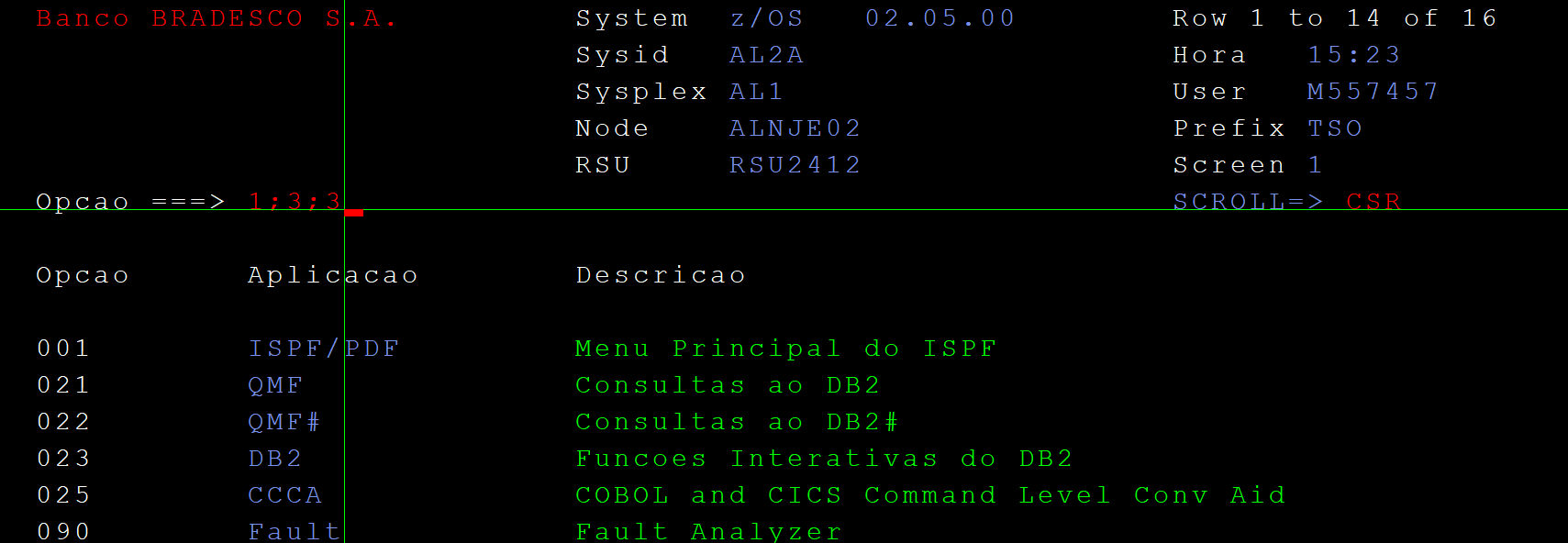
# Introdução

Neste runbook, veremos realizar testes unitários scripts/tabelas Teradata, mais especificamente na camada Stage. Para isso, são necessários alguns pré-requisitos:

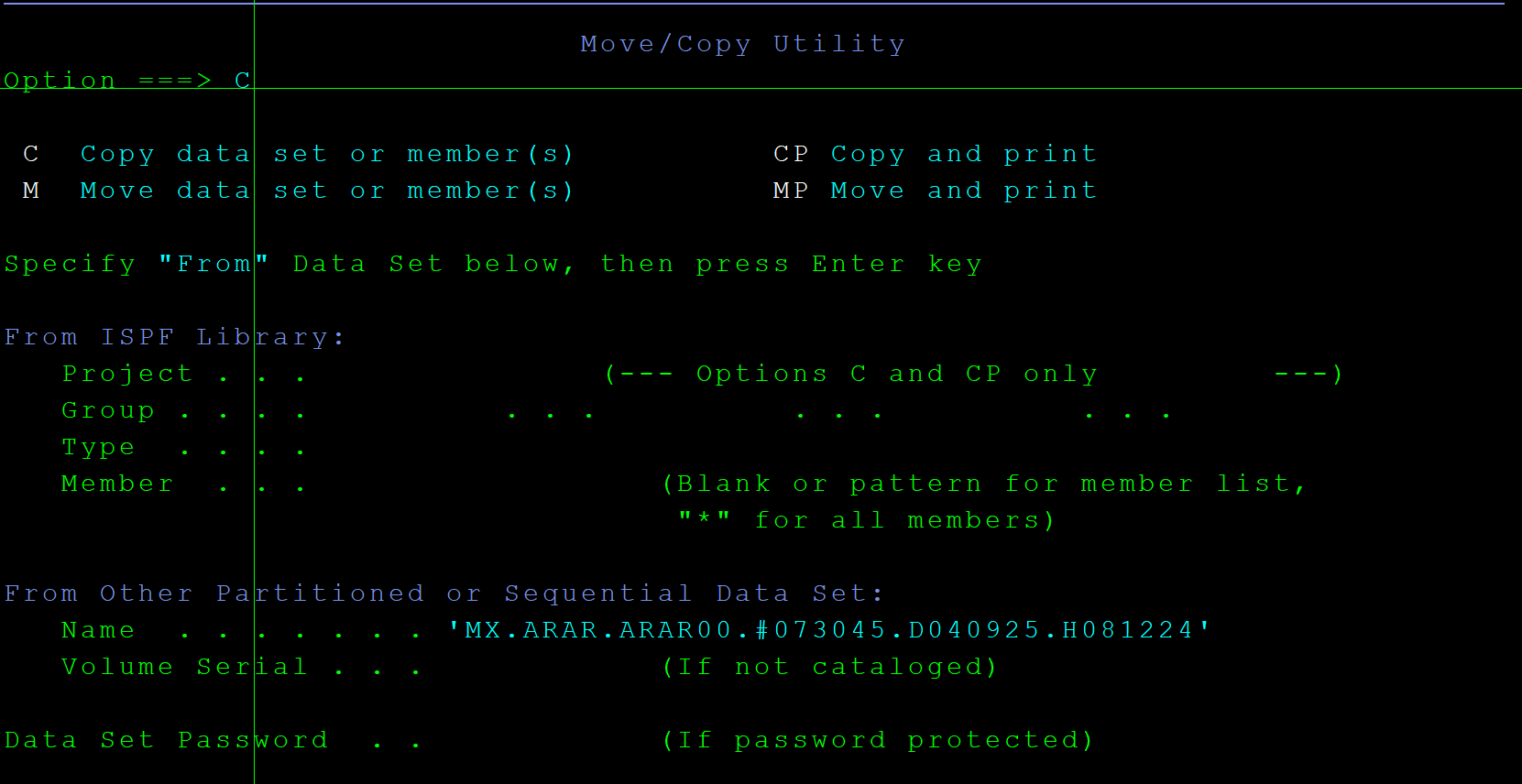
* Package criada com o script deseja já alterado (caso não saiba como fazer isso, consulte o runbook [Implatação de Scripts Teradata no Mainframe: from Hero do Zero](https://github.com/KaelDucatti/tdw-migracao-cpf-cnpj-bradesco/blob/main/implantacao-from-hero-to-zero.pdf).
* Tabela Teradata modificada, para atender às modificações feitas no script do Teradata.
* Massa de dados referente a tabela alterada no ambiente onde está sendo executado o teste (o pedido é feito pelo HexaVision, na aba de serviços).
* Conhecimento básico sobre mainframe (consulte o runbook [TSO: amigo ou inimigo?](https://github.com/KaelDucatti/tdw-migracao-cpf-cnpj-bradesco/blob/main/tso-amigo-ou-inimigo.pdf)).
* Por fim, é preciso ter paciência...

## Como copiar a Massa de dados para um arquivo em uma outra pasta nossa.

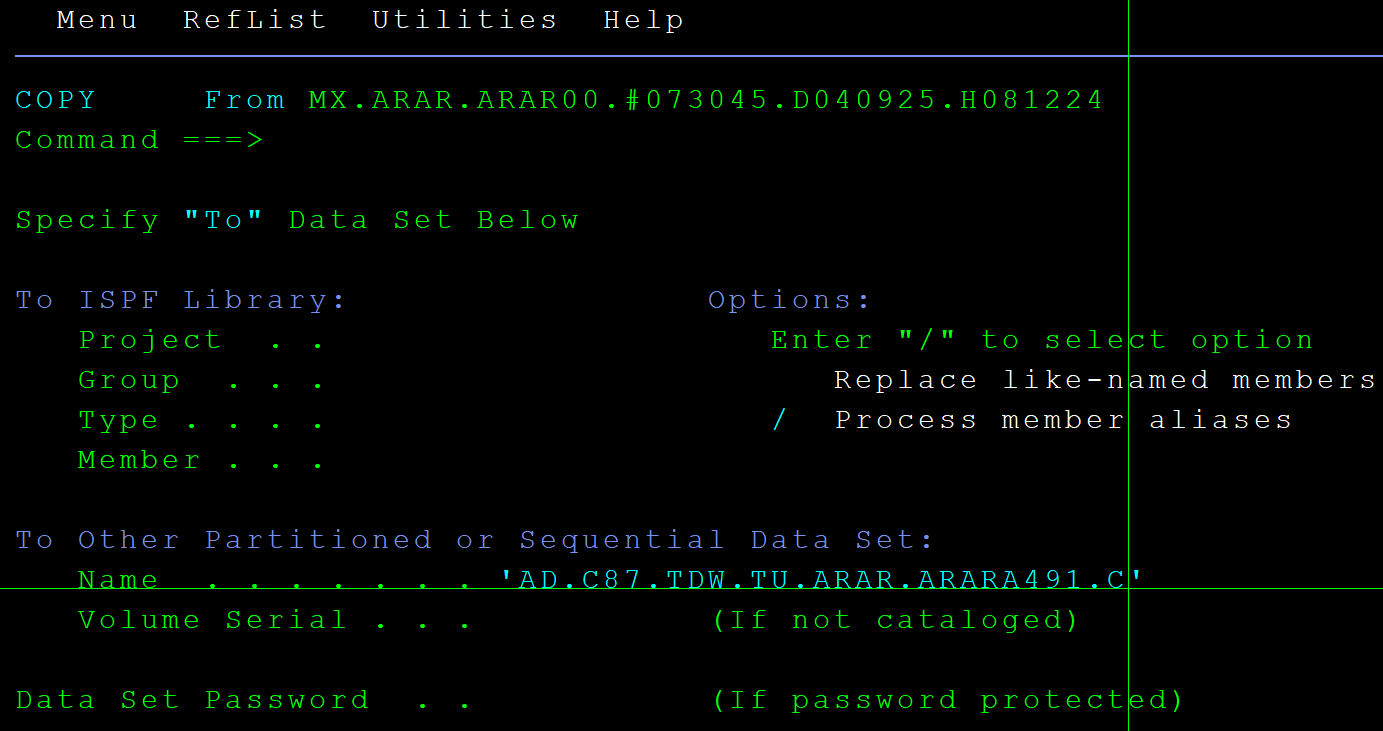
1. Entre no menu 1;3;3



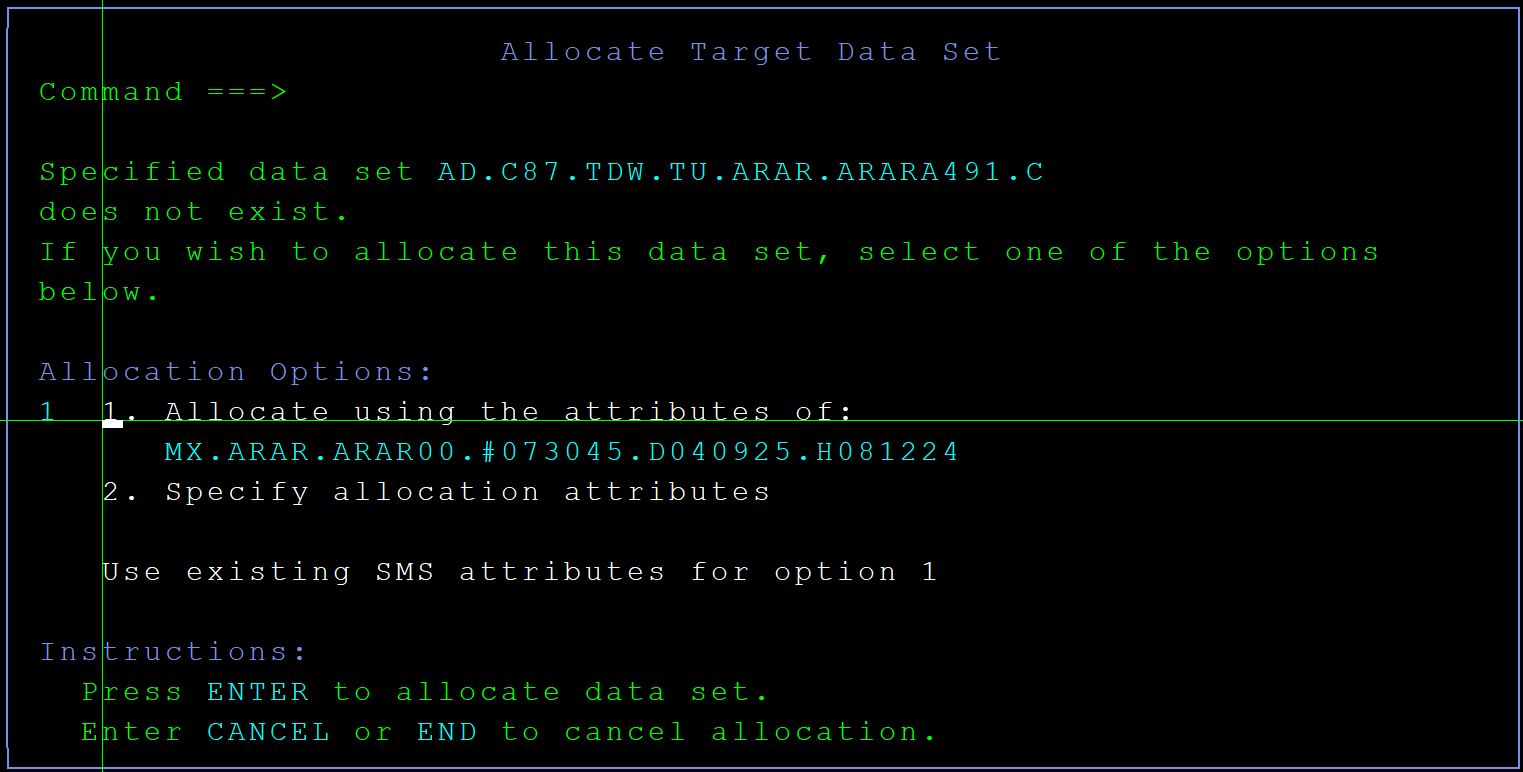
1. Informe a fonte a ser copiada e seleciona a opção C (copy).



1. Insira o local de destino como o nome que o arquivo que deseja que o arquivo tenha.



1. Na próxima tela que abrir, insira o número 1 pressione a tecla Enter.



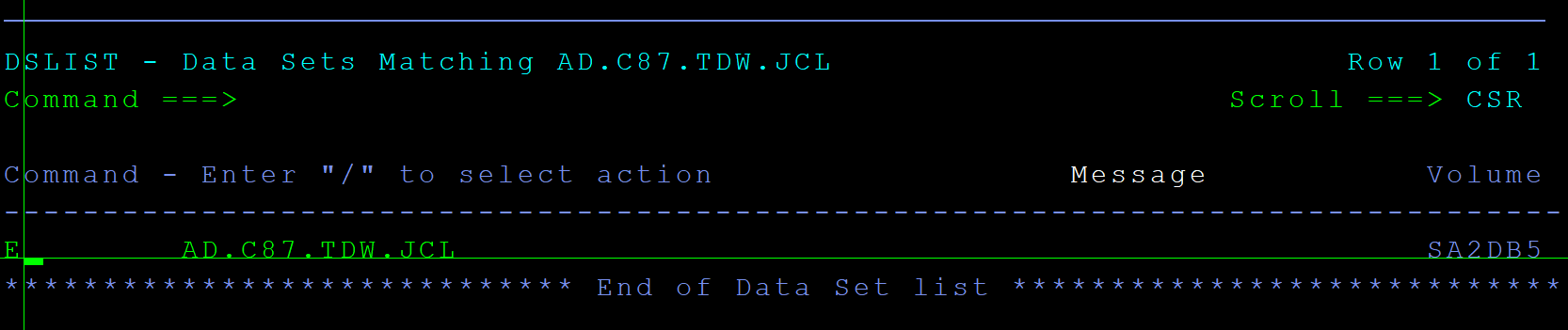
1. Feito! Dataset copiado com sucesso para a pasta destino!



## Como criar o nosso segundo arquivo de teste para o cenário A (com os campos extras na tabela preenchidos).

1. Antes de tudo, precisamos já ter nosso arquivo de teste para o plano B feito. Depois disso, iremos alterar alguns campos em um determinado job para que o mesmo faça uma cópia dessa massa de dados com os campos extras. Comecemos copiando esse arquivo para que consigamos nos organizar!

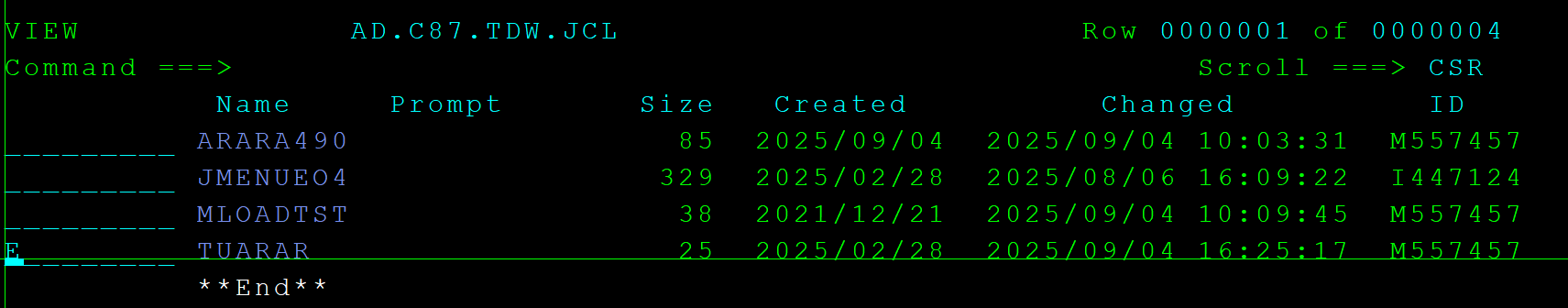
No menu 1;3;4, entre na pasta AD.C87.TDW.JCL.



1. Nessa pasta iremos encontrar todos os nossos arquivos JCL. Dentre esses arquivos, encontramos principalmente 2 tipos:

* Os que tem o nome de um scripts: utilizados para executar o teste unitário no script que levam o nome.
* Os que começam com TU (Teste Unitários) e TI (Teste Integrados), utilizados para gerar a massa de dados que é utilizada para testar o plano A.

Por hora queremos ver o que tem como sufixo TU, pois é o foco deste tópico.



1. Nesse job, o que nos interessa é o que tem a partir do STEP01. Aqui temos 3 locais importantes:

* **SORTIN:** nesse campo precisamos inserir a massa de dados que copiamos para o teste do plano B (a que não tem os campos extras).
* **SORTOUT:** esse é o nome do arquivo que iremos gerar para o teste do plano A. É de boa prática que sigamos o sufixo presente, modificando apenas o centro de custo (ARAR), nome do script (ARARA491) e plano (A), para o que se adequa ao seu caso.
* **OUTREC BUILD:** define como os dados do SORTIN serão formatados no SORTOUT. Embora o conteúdo original esteja disponível, ele só será efetivamente estruturado com o uso do OUTREC.

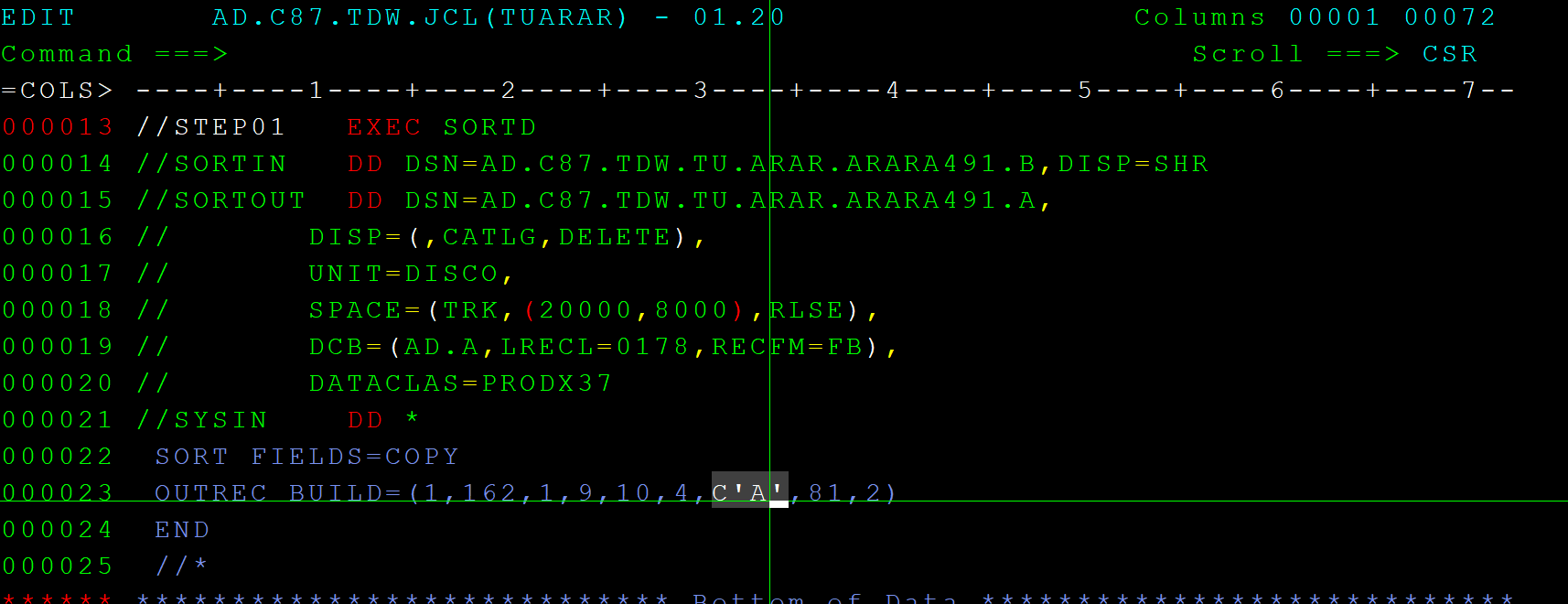
Cada par de valores representa:

* + **posição inicial** da coluna a copiar,
  + e **quantidade de bytes** a extrair.

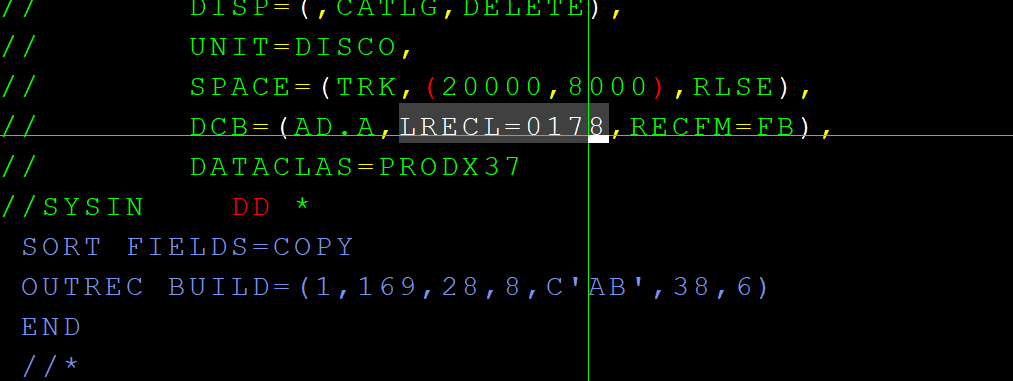
Por exemplo:

* (1,162) copia os primeiros 162
* (1,9) extrai os 9 primeiros bytes como campos adicionais.
* (10,4) pega 4 bytes a partir da posição 10.
* C'A' insere a letra A em todas as linhas nessa posição bytes do registro original.

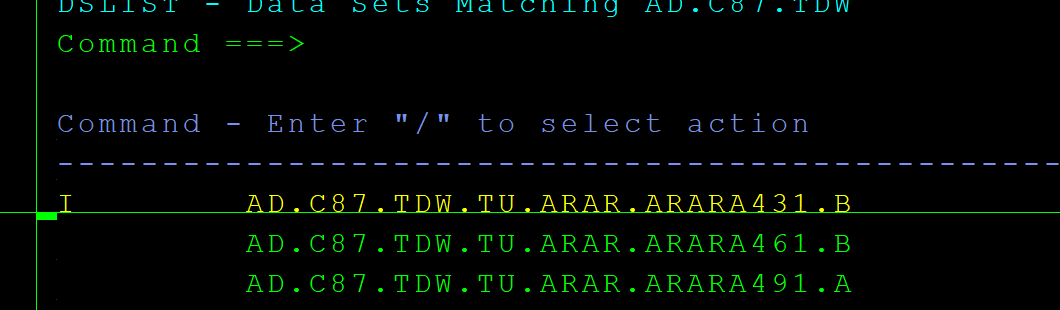
Esse processo permite montar uma nova estrutura de dados conforme a necessidade.

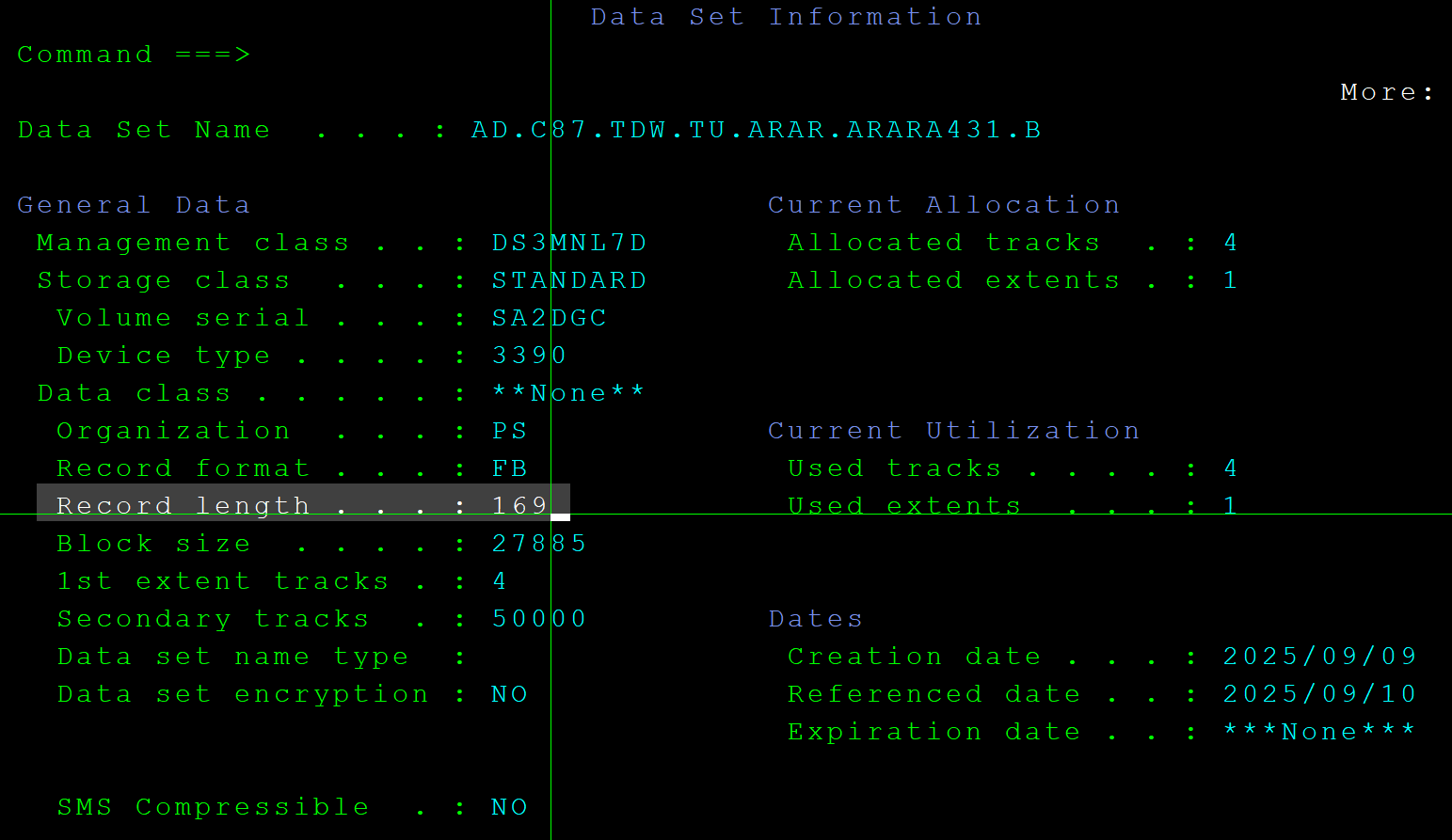


* **LRECL:** Ok. Agora precisamos mudar o valor dessa variável também. Iremos atualizá-la com o novo tamanho total do nosso arquivo. Se antes ele tinha 162 bytes e adicionamos 16 colunas nele, agora ele terá 178 bytes (1 bytes = 1 coluna).

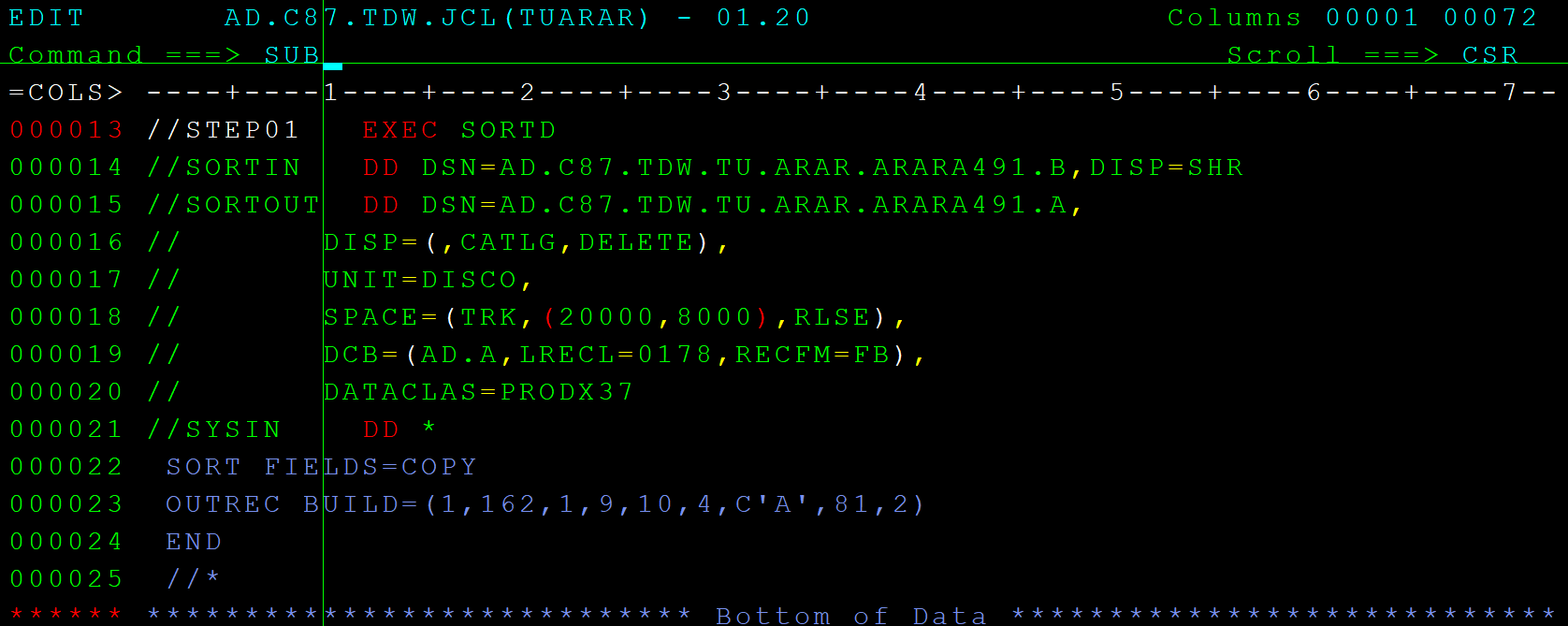


Dica: Se você inserir a letra I em frente ao arquivo que quer ver e pressionar Enter, irá para uma tela de informações daquele arquivo. Nessa tela, você consegue descobrir várias coisas, inclusive o tamanho total do arquivo no campo “Record length” (o que nos é muito útil).





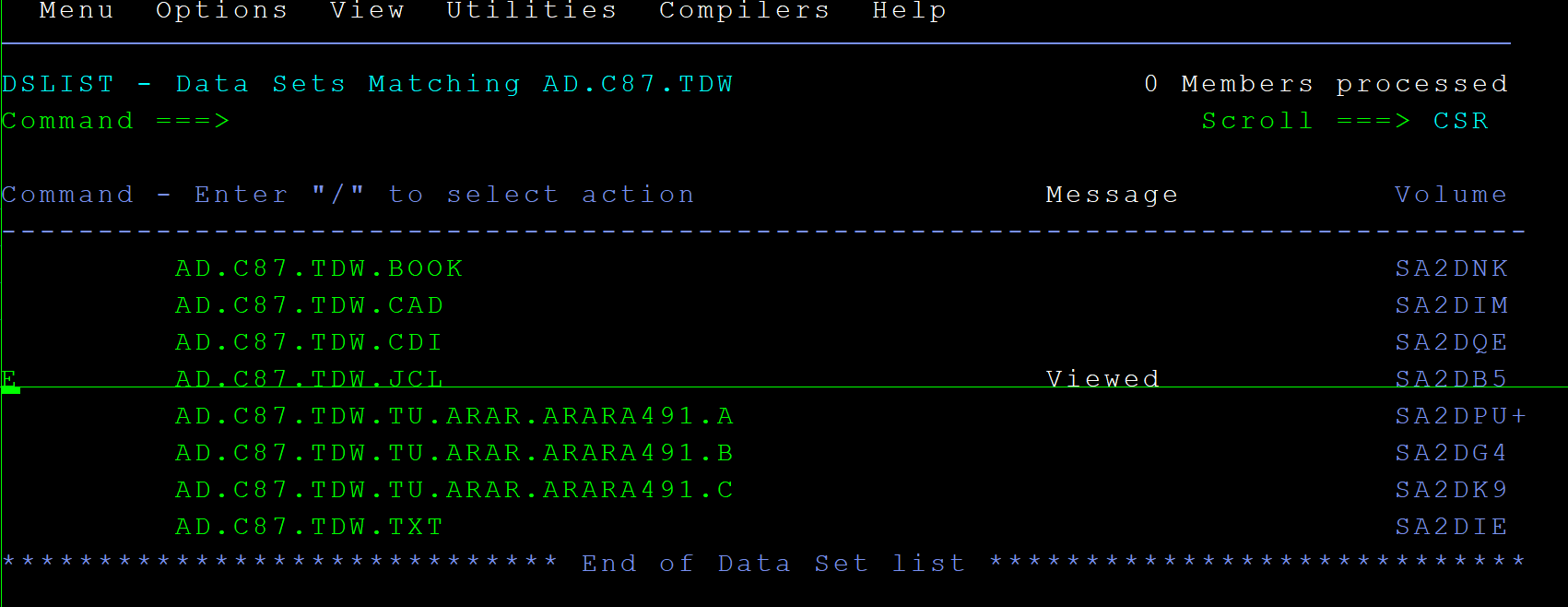
1. Alterado nosso JOB, podemos submetê-lo com comando SUB.



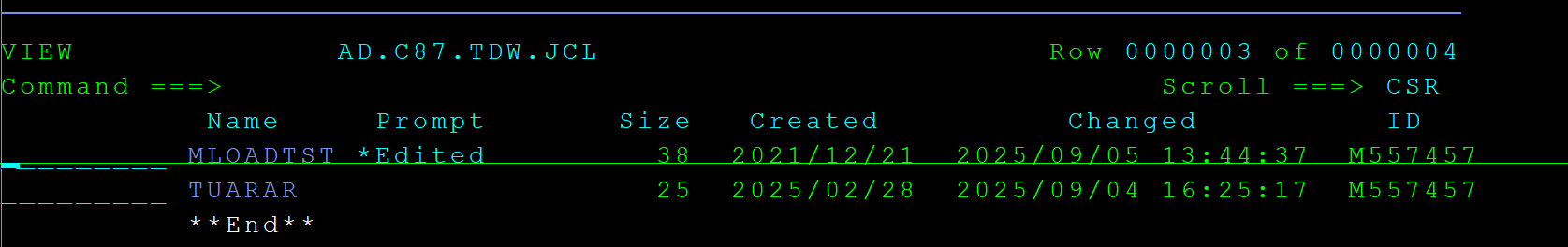
1. Feito isso, é esperado que seu arquivo seja criado com sucesso. E você tenha enfim as duas massas de dados para realizar seus testes de plano A e B.

## Como executar os testes unitários para os planos A e B?

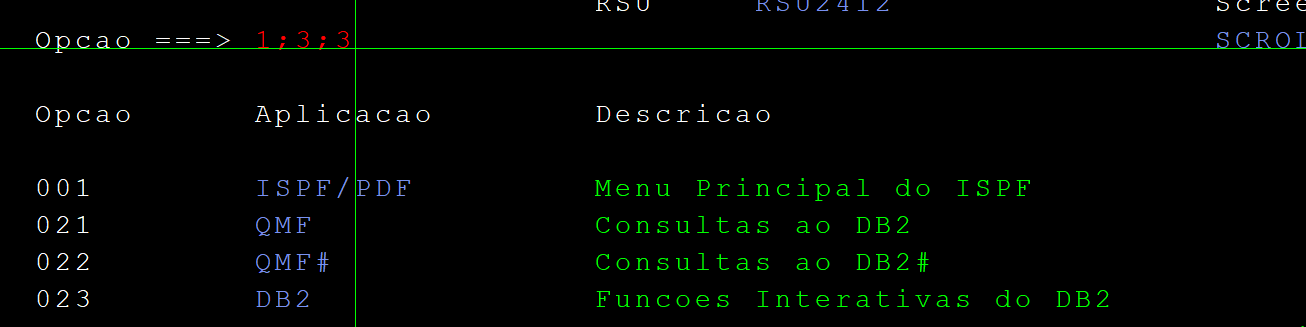
1. Hora de brincar com o Job novamente!  
   Mas calma, dessa vez o objetivo é outro: vamos fazer nosso script do Teradata rodar com as massas de dados que geramos.  
   Então, respira fundo, abre o ISPF e segue o caminho mágico: 1;3;4. Lá dentro, mergulhe na pasta **AD.C87.TDW.JCL** — o lar dos nossos Jobs.



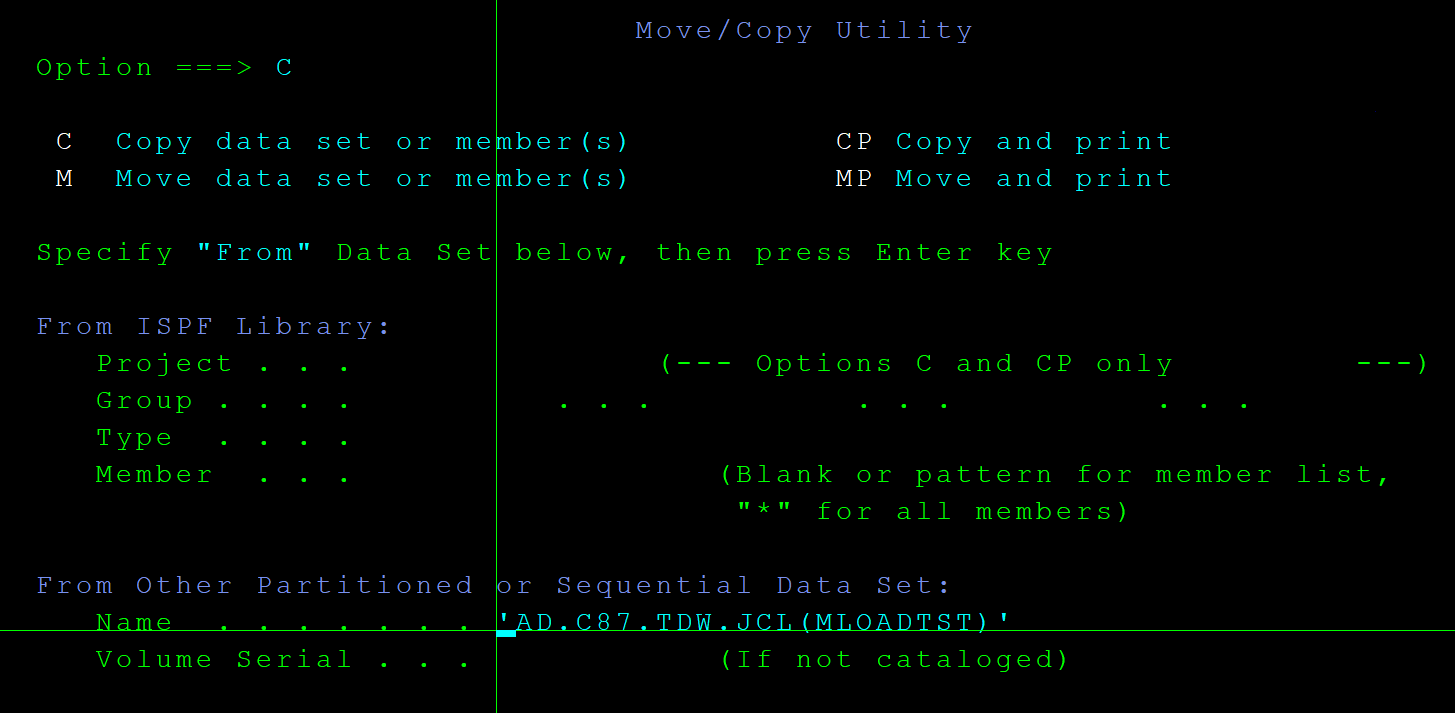
1. Aqui dentro queremos vamos querer fazer uma cópia do Job MLOADTST. Esse arquivo é o Job base para executarmos nossos testes.



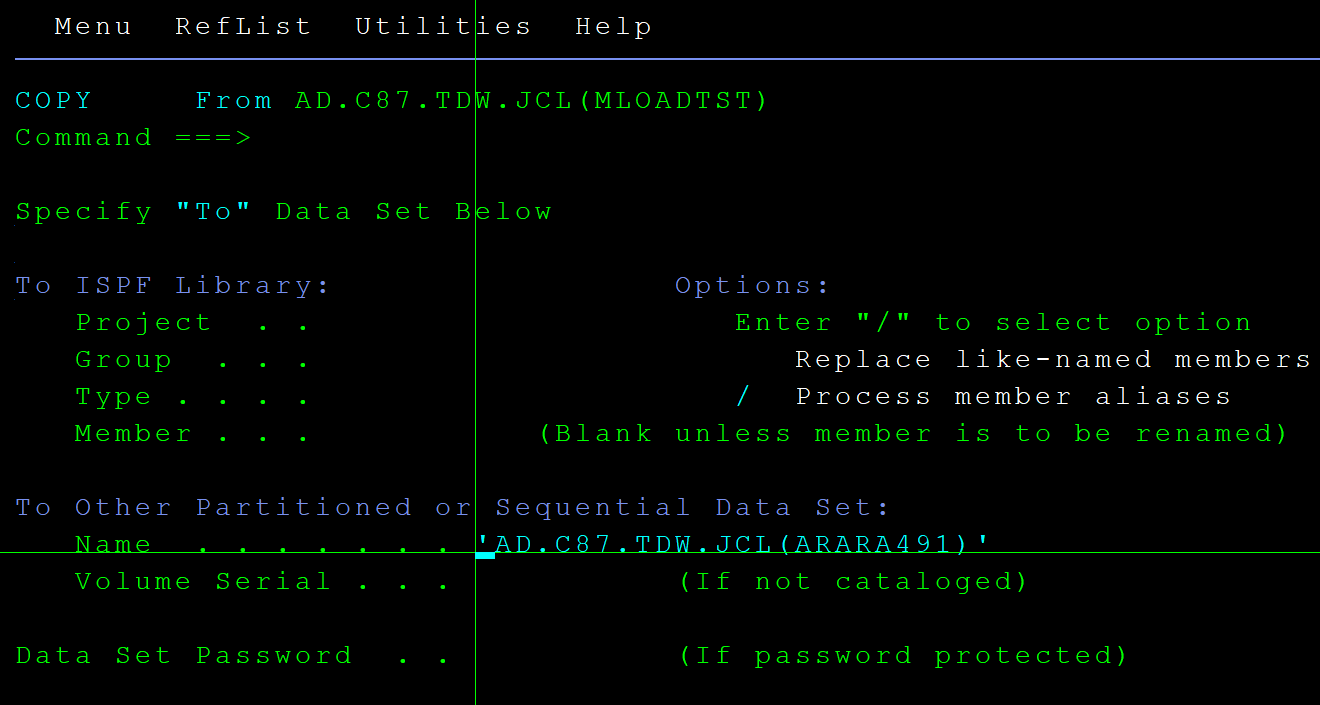
1. Para copiarmos esse job, iremos utilizar o menu 1;3;3.



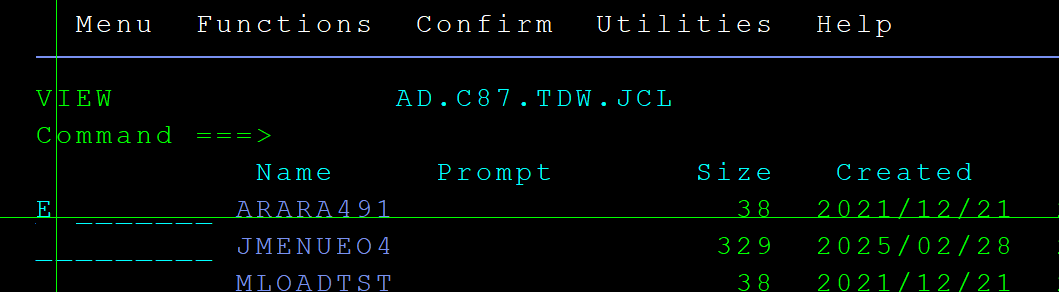
1. Aqui inserimos a opção C (copy) no campo command e em name o path para o arquivo que queremos copiar.



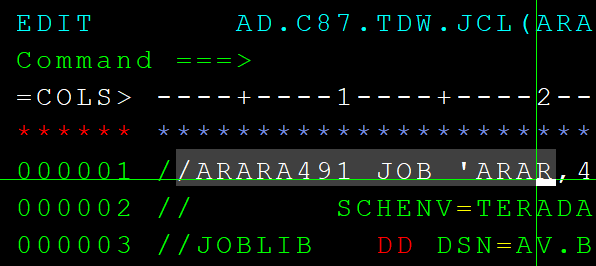
1. Agora inserimos o path para onde queremos copiá-lo já com nome que desejamos que ele tenha no destino.



1. Pronto! Podemos voltar ao AD.C87.TDW.JCL e entrar nesse Job para modifica-lo ao nosso bel prazer.

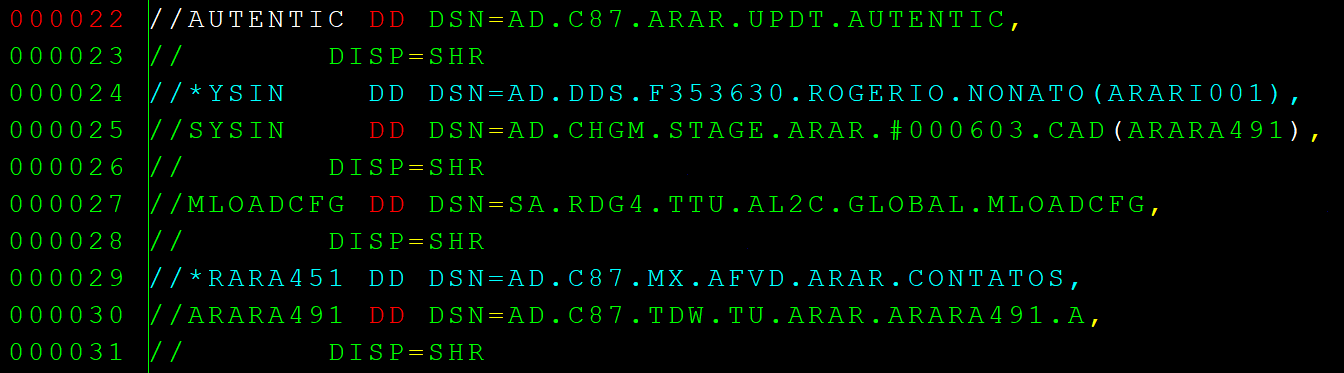


1. A primeira coisa que iremos mudar nesse Job está logo na primeira linha. Primeiro renomeamos o nome do Job para o nome do script que estamos testanto (por organização) e depois o nome do centro de custo ao qual ele está relacionado (no caso, ARAR).



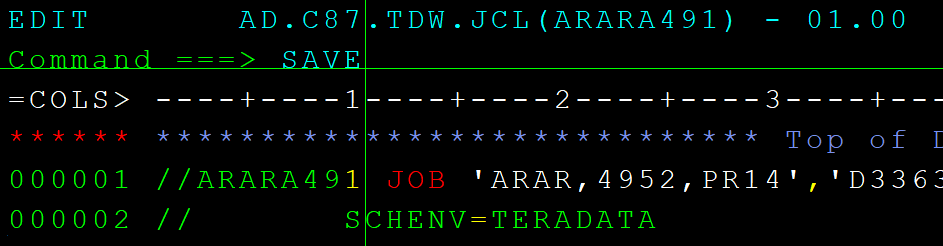
1. Na sequência, iremos lá para onde começa o STEP2, onde executamos o MLOAD. Temos dois locais importantes para serem observados:

* **SYSIN:** Aqui é onde apontamos para o script Teradata que você já deu aquele tapa de mestre(a). Ele está guardado com carinho dentro da nossa package, esperando ser chamado para o palco.
* **ARARA491:** Esse nome exótico não é um código secreto, mas sim o lugar onde nossa massa de dados para teste está repousando. E atenção: nessa linha não mudamos só o conteúdo atribuído à variável DSN, mas também o prefixo “ARARA491” — que precisa ser trocado pelo nome do script que estamos executando.

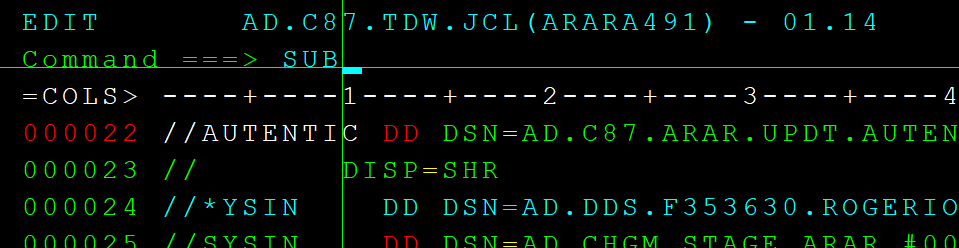


Obs.: as linhas com um asterisco (\*) na frente são consideradas comentários.

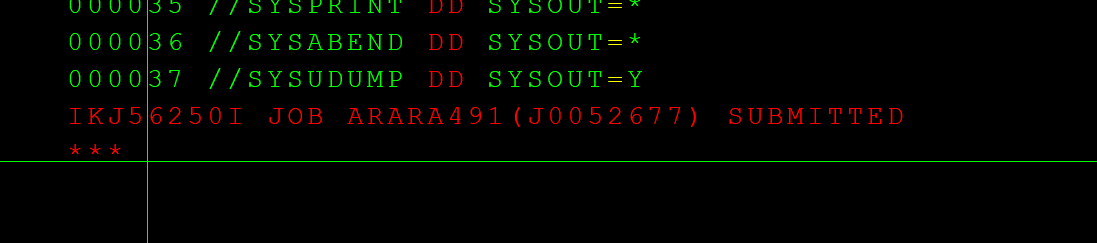
1. Feitas as mudanças, agora salve o arquivo.



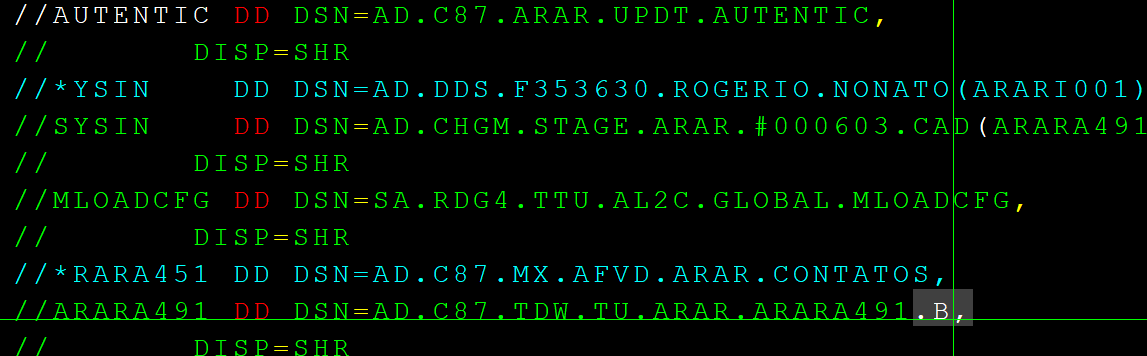
1. E submeta ele.



1. Após submeter, você verá essa mensagem vermelha no canto inferior esquerdo da sua sessão. Não se assuste, significa que sua chamada ao Job foi executada.

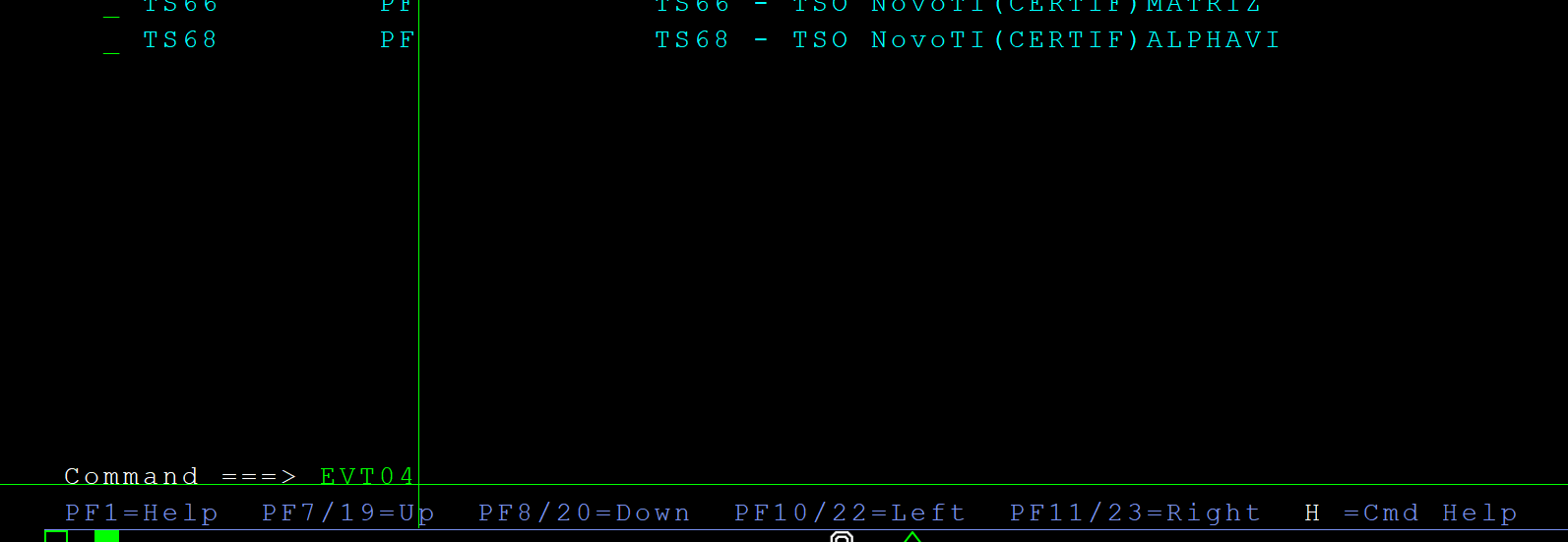


1. Faça o mesmo processo, mas agora com a massa de dados do plano B (sem as colunas extras).

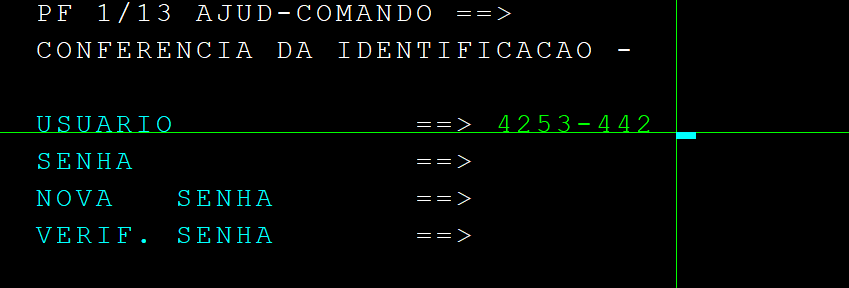


## Rodamos o Job de teste com a massa de dados, mas será que ele executou corretamente?

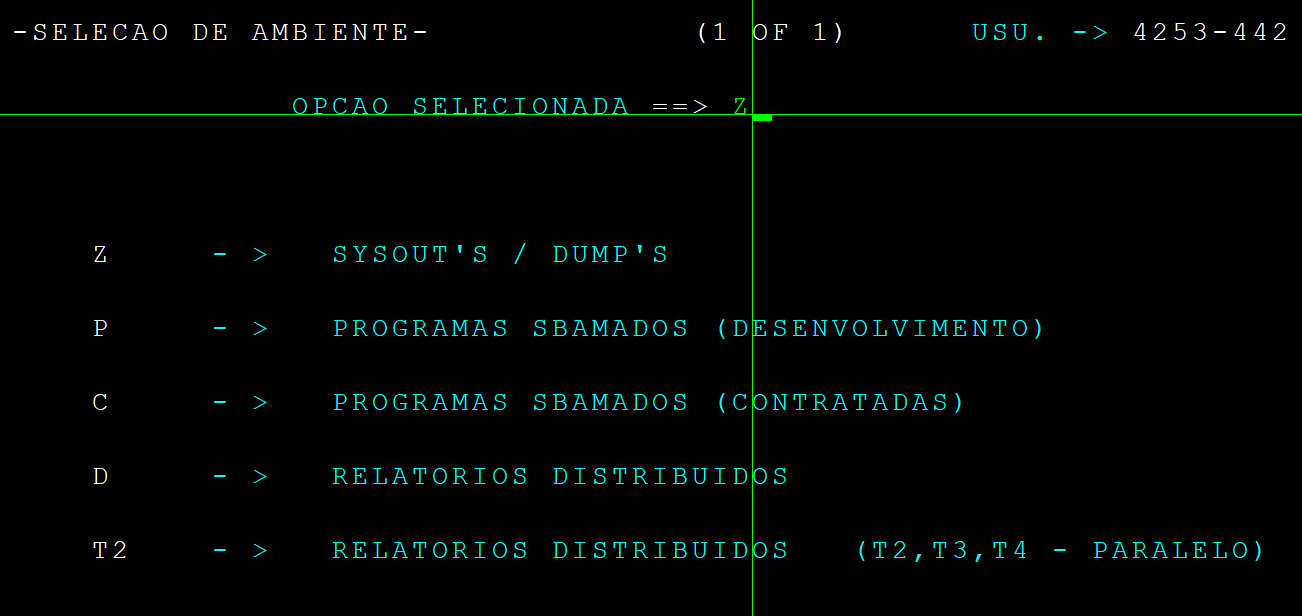
1. Primeiro local que iremos utilizar para verificar se o Job rodou é o EVT. Para isso, saia do ambiente em que você está (provavelmente o TS04) e entre no EVT04 (sim, tem um EVT para cada ambiente).



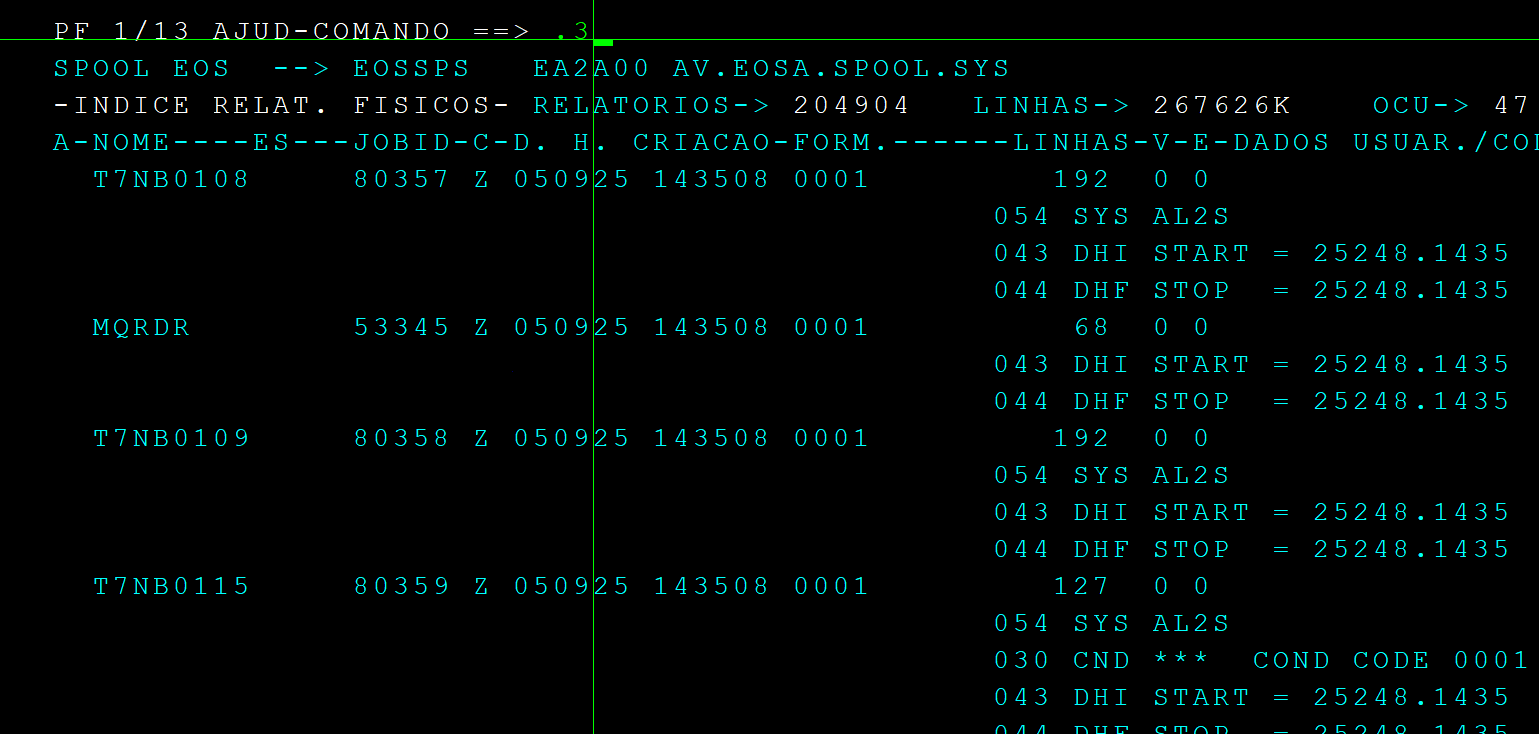
1. No campo usuário, insira o valor **4253-442** e pressione Enter.

****

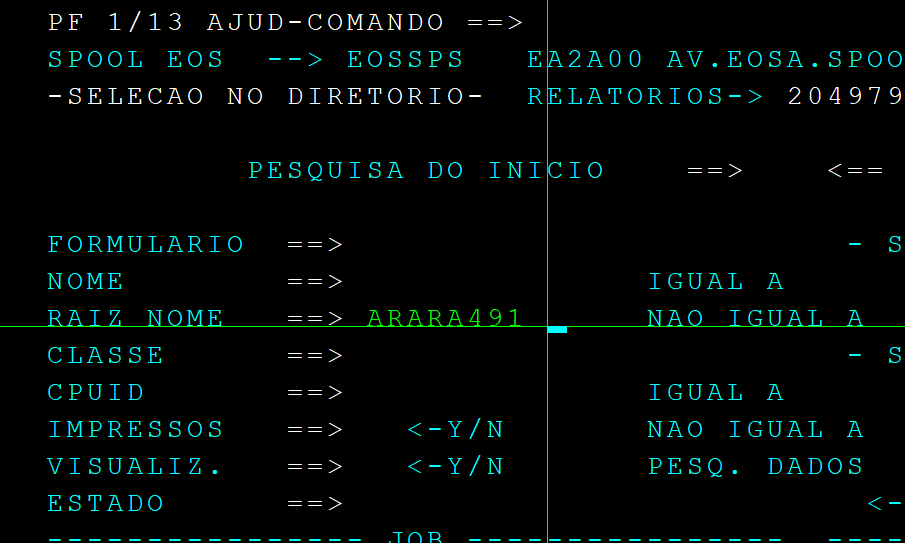
1. Selecione a opção Z.

****

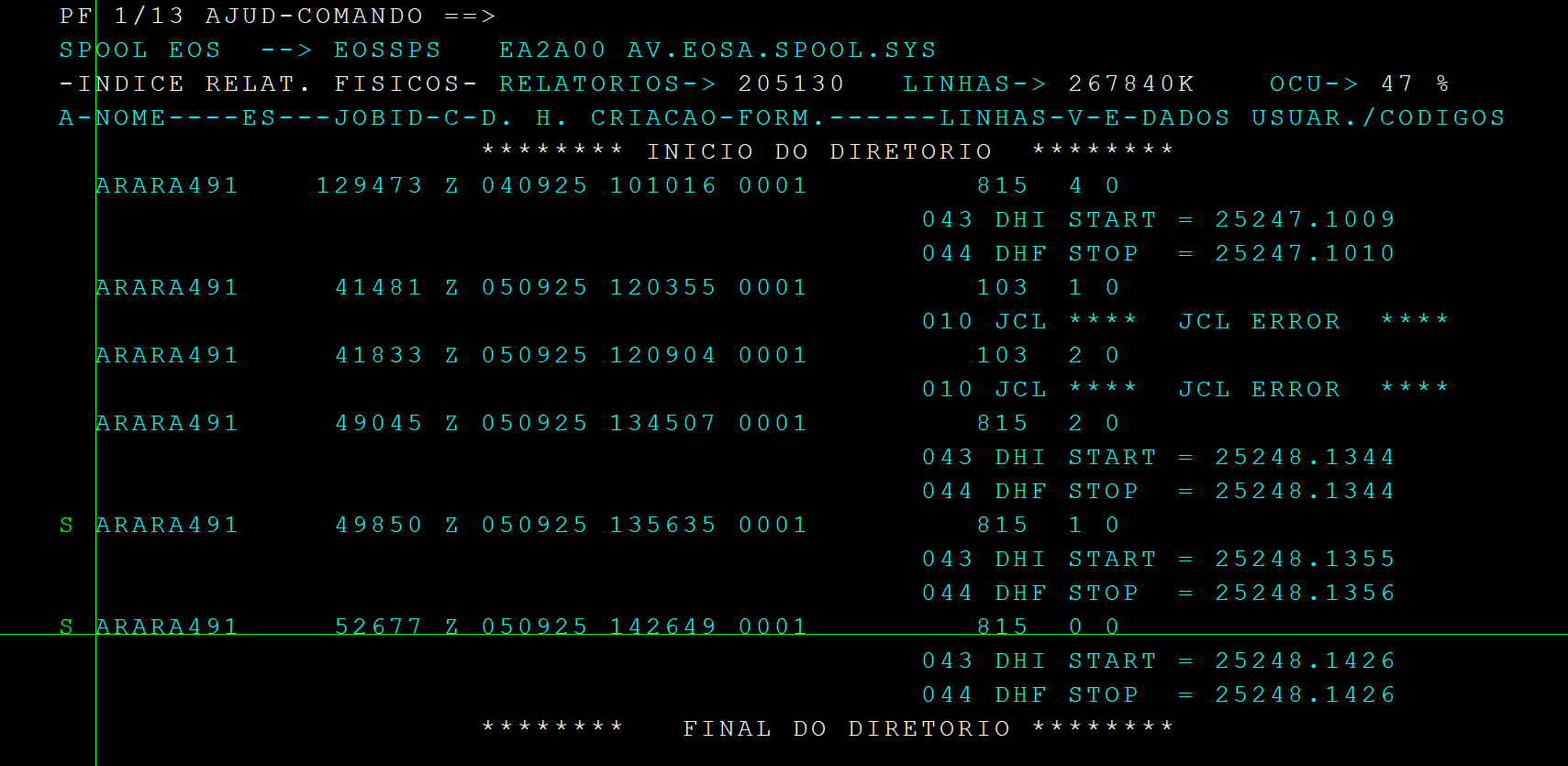
1. Agora veremos o histórico de tudo que foi rodado a pouco tempo no TS04, mas ainda não é aqui que precisamos estar. Insira o comando **.3**.

****

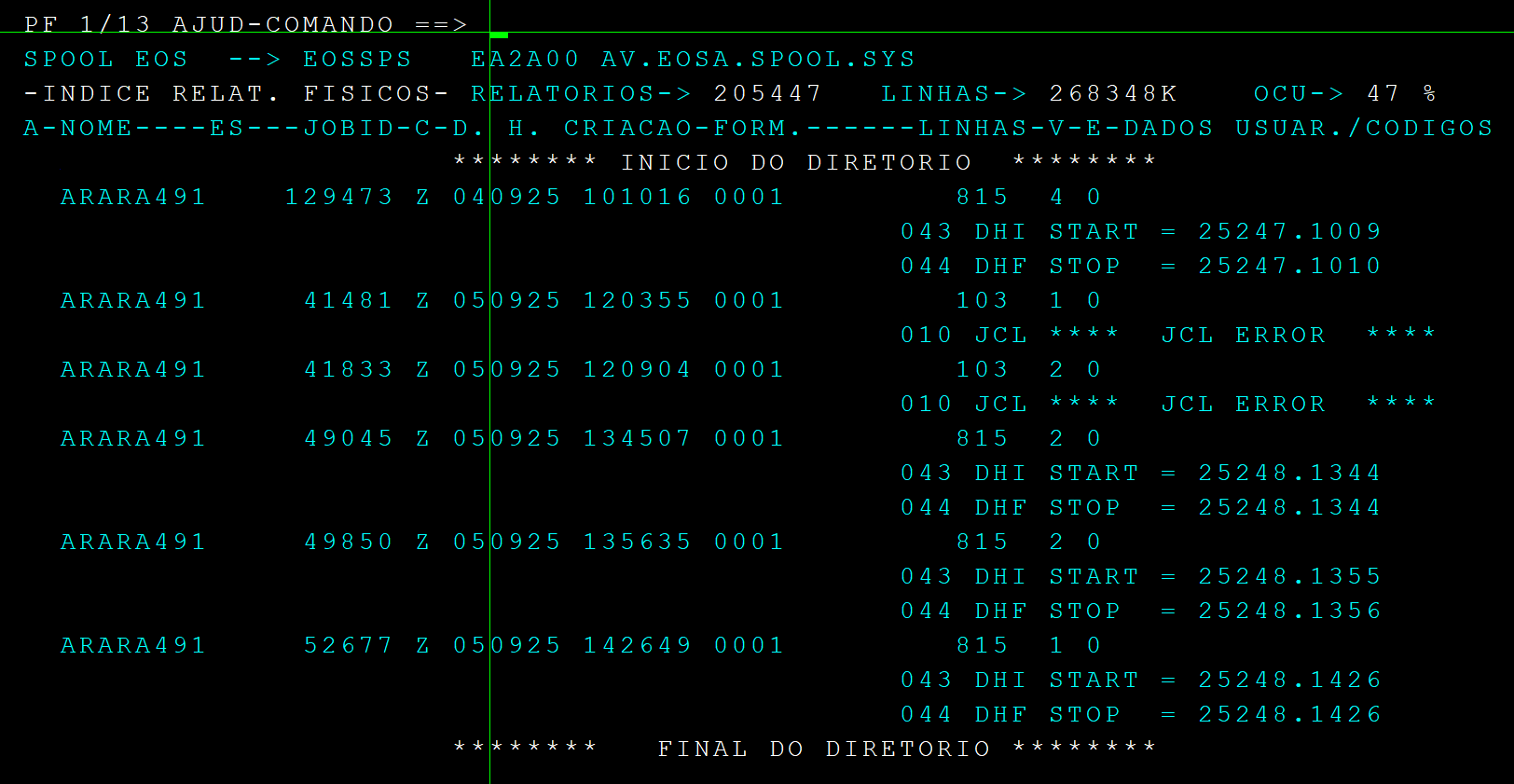
1. Pronto, agora sim. Insira no campo “RAIZ NOME” o nome que você escolhe para o Job no passo 7do tópico anterior, onde aprendemos a como executar um Job.

****

1. Agora sim temos a lista que queremos ver. A última linha é o Job mais recente, a primeira, o mais antigo (eu sei, não faz sentido). Se você seguiu o passo a passo do teste corretamente, é para essa lista ter pelo menos o histórico de 2 Jobs executados: o pro plano A e o pro plano B. Digite **S** na frente deles.

****

1. A primeira informação que temos aqui é que o Job foi realmente executado. Se notar nos Jobs mais acima, há uma mensagem com “JCL ERROR”, enquanto em outros há apenas a momento de START e STOP.

****

1. Agora, já dentro do Job, podemos fazer a coleta de evidências. Pode ser que seja necessário utilizar o Teradata Studio também e fazer um **SELECT \* FROM tabelaX;** para cada um dos testes (A e B), de maneira a verificar o comportamento da tabela em cada situação.

As evidências dependem de ambiente para ambiente, então alinhe com a sua equipe para entender o que é necessário para comprovar a execução efetiva do teste.

Uau! Você chegou até aqui!

Parabéns! Você é oficialmente um(a)

**Mainframe Data Engineer**

