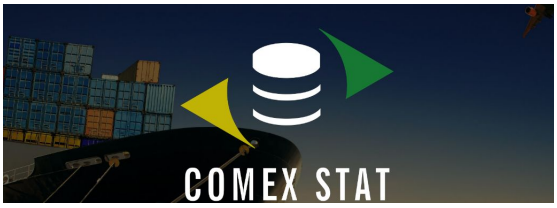
Projeto Final Data Science - Harve

O projeto final do curso de Data Science foi o meu primeiro projeto de Ciência de Dados. Eu quis unir os novos conhecimentos adquiridos durante o curso com os meus conhecimentos profissionais. Sou formada em administração de empresas e trabalho com Importações e Exportações a muitos anos.

Eu quis aproveitar a oportunidade para montar um projeto do zero, ver como seria de A a Z. O meu projeto final teve como objetivo analisar alguns dados brutos do **Comex Stat** utilizando serviço em nuvem da Amazon - **AWS.**

O Comex Stat é o portal gratuito para consultas às estatísticas de comércio exterior



do Brasil, do Ministério da Economia. São divulgados mensalmente os dados detalhados das **exportações e importações** brasileiras, baseados na declaração dos exportadores e importadores no SISCOMEX (Sistema Integrado de Comércio Exterior).

A Amazon Web Services (AWS) existe desde 2006, oferece **mais de 175 serviços** completos de datacenters para empresas por meio da computação em nuvem. Segundo a própria AWS essa infraestrutura de nuvem global é a mais segura, abrangente e confiável plataforma existente. A AWS **tem como objetivo reduzir custos e prover escala computacional.**

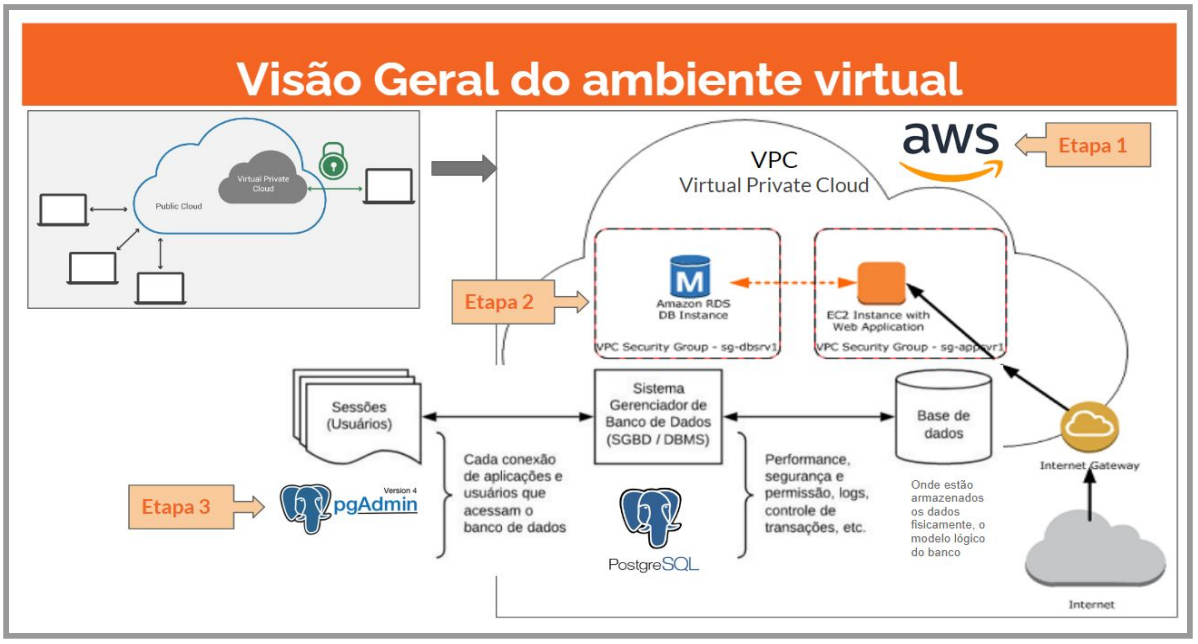
Você escolhe os serviços que quer usar e como vai usar, pagando só pelo que usa e encerrando quando não precisar mais. Então o custo-benefício, a escalabilidade e a elasticidade são os seus pontos fortes. Esses serviços ajudam as empresas a serem **mais produtivas e a focarem no seu core business.** É utilizada por centenas de milhares de empresas em 190 países no mundo.

ETAPAS DO PROJETO

O projeto foi composto por várias etapas, que vou detalhar ao longo da matéria:

1. Criar conta na Amazon Web Services - AWS
2. Criar Database na Amazon RDS – Relational Database Service
3. Conectar-se a uma instância de banco de dados PostgreSQL
4. Quais arquivos serão analisados?
5. PgAdmin 4 - “Importar arquivo”
6. Análise dos dados das tabelas
7. Visualização com Power BI

Então vamos lá, vou começar pela visão geral do ambiente virtual.



O que é um **VPC**?
- Virtual Private Cloud é a criação de uma nuvem privada dentro da nuvem pública, a qual é utilizada na estrutura de funcionamento da AWS. Um VPC é criado no momento em que você cria uma conta na AWS, que foi a **Primeira Etapa** do projeto.

1. **Criar conta na Amazon Web Services - AWS**

Criar a conta na AWS é fácil! Basta seguir as instruções do site. Será necessário indicar o número de um cartão de crédito internacional*(*)*, para que seus dados sejam autenticados e para garantir, que a Amazon, possa receber por seus serviços. Nesse caso é debitado o valor de US$1,00 no cartão de crédito para validação. Após isso é enviado um código de verificação via SMS, para concluir o cadastro. E Etapa Concluída!! Criada a conta Raiz!!! **Ao se cadastrar, você tem acesso automático a todos os serviços da AWS.**

(*) A partir de 1º de novembro de 2020, a AWS lançará a Amazon AWS Serviços Brasil Ltda. (AWS SBL) para atuar no Brasil, como entidade local para prestação de serviços AWS e faturamento para clientes brasileiros. Ou seja, passará a aceitar outras formas de pagamento, inclusive através de cartão de crédito nacional.

A documentação do AWS é bem ampla, explica passo a passo de como criar e utilizar os serviços disponíveis. Além de sugestões de boas práticas para utilização. Exemplo disso é a criação do Usuário Administrador no IAM - Identity and Access Management. Depois você adiciona o usuário a um grupo de administradores, o que possibilita criar grupos com vários usuários e senhas individuais, com permissões específicas. Como o professor Charles criou para nós na aula de SQL.

2. **Criar Database na Amazon RDS**

Segunda Etapa - Mas o que é o Amazon RDS ?
– **Relational Database Service (RDS)** é o **Sistema Gerenciador de Banco de dados**. A função dele é a performance; segurança e permissão; logs; controle de transações; entre outras coisas.

O Amazon RDS permite a fácil configuração, operação e escalabilidade de bancos de dados relacionais em cloud computing. O recurso proporciona capacidade redimensionável e econômica, além de **automatizar tarefas de administração complexas**, como: configuração de bancos de dados; provisionamento de hardware; backups; aplicação de patches. Assim, você trabalha de maneira mais segura e de acordo com as conformidades que as aplicações necessitam.

O RDS oferece seis diferentes tipos (versões ou "sabores") de banco de dados, que são:

1. PostgreSQL;
2. Amazon Aurora;
3. MySQL;
4. Microsoft SQL Server;
5. Oracle;
6. MariaD.

O Amazon RDS cria automaticamente (sem que você perceba) um local para que o banco de dados seja armazenado fisicamente, o Amazon EC2, o modelo lógico do banco.

O Primeiro Passo após entrar na conta Administrador/ Raíz é escolher a região da AWS na qual você deseja criar a instância de banco de dados.A sua conta determina as regiões que estarão disponíveis para você. Você pode utilizar uma mais próxima geograficamente (como eu fiz) ou escolher outra qualquer.

A AWS abrange 77 Zonas de disponibilidade em 24 regiões geográficas por todo o mundo e já anunciou planos de expansão.

Depois vá em 'Criar banco de dados". Siga os passos, é bem intuitivo. O indicado é utilizar o método **"Easy Create"**. Para o meu projeto escolhi as configurações **PostgreSQL** e Free tier (gratuita). Crie um nome para o banco de dados, além do usuário master e senha.

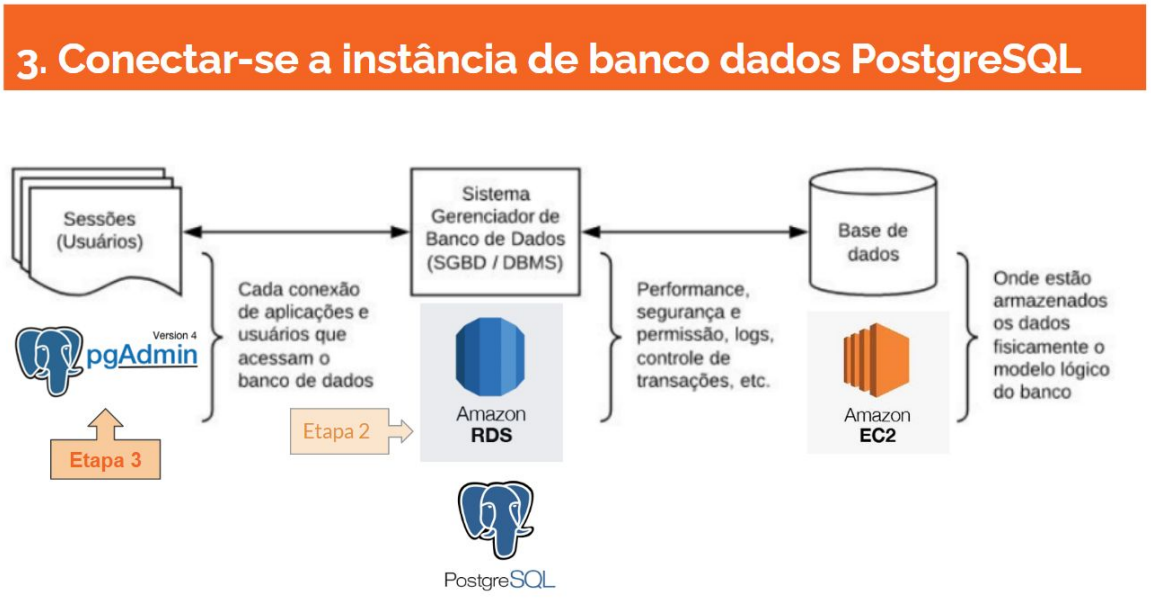
E por fim vá em "Criar Database". Demora um pouco, dependendo do tipo de computador e Internet que se está utilizando. Então finalmente aparece uma nova tela indicando que o banco de dados foi criado e está disponível.

É importante destacar, que ao clicar no banco de dados criado, você encontra o resumo dele, ou seja os detalhes: região onde está, tamanho, versão utilizada e disponibilidade.

E na aba “Conectividade e Segurança” você encontra entre outras coisas:

- ➔ **Endpoint :** é o endereço para acessar o banco de dados depois.
- ➔ **VPC** : endereço da sua nuvem privada
- ➔ **VPC Grupos de Segurança :** regras de entrada e saída

3. Conectar-se a uma instância de banco de dados PostgreSQL



O próximo passo é a **conexão** da instância de banco de dados do PostgreSQL, eu utilizei o programa **PgAdmin**, versão 4. Foi basicamente como vimos na aula de SQL. Cria-se um novo server, que será direcionado para a minha base de dados criada anteriormente no Amazon RDS.

Então, vai em criar novo server e depois coloca um nome pra ele e o Endpoint (endereço) e porta indicados no Amazon RDS. Além do usuário e senha master criados lá. E vai em criar. E fica na expectativa do que vai acontecer, porque a hora da verdade chegou!!

Após alguns segundos vem o resultado:

Erro : Incapaz de se conectar com o servidor

Tempo limite expirado

Não era a resposta esperada! É o momento que você xinga, se desespera, quer bater no computador. E começa a pensar: “Eu fiz tudo certinho, segui todos os passos como na documentação! Caramba!! Primeira tentativa e deu erro. O que eu fiz de errado?”. “Mas calma, não vou desistir, vamos resolver!”

Volto à etapa 2, no Amazon RDS, para identificar o motivo deste erro. Releio toda a documentação, tento pensar porque este erro aparece e com algumas possibilidades e muitas dúvidas na cabeça, marco uma mentoria com o professor Charles, para tentar entender o que aconteceu.

E concluímos que foi um caso literal de “bater com o nariz na porta” ou melhor fui “barrada no baile”. **O erro ocorre por dois motivos: Regra de entrada no grupo de segurança e Acesso Público.**

- **Primeiro motivo**: É necessário colocar uma regra específica para o **PostgreSql** com origem geral no grupo de segurança. Para que qualquer pessoa que tenha o Endpoint do servidor possa acessá-lo.

- **Segundo e crucial motivo** : O acesso deve estar como público e não restrito, como ocorre quando você cria o Database utilizando o Easy create.
    - Para resolver, a maneira mais fácil foi deletar o banco de dados e depois restaurá-lo. Assim ele pergunta se deve ser público ou não. Coloca como sim. E resolvido!!

Mas ninguém conta isso pra você, não aparece na documentação claramente. Na página para criar o Database, no Amazon RDS, é possível ver a configuração padrão ao utilizar o Easy Create. Lá aparece como padrão “Acesso Público - Não”. É possível alterá-lo posteriormente, mas de uma forma mais difícil.

Depois de alterar os dois itens, volto à **etapa 3 de criação de server**, coloco novamente os dados: nome para o server; Host; usuário e senha master e vou em criar.

E finalmente conecta!!! Viva, consegui!!
Foi um momento de muita emoção, kkk!! Consegui conectar ao banco de dados que eu mesma tinha criado na nuvem!!! E sem ser de TI, que felicidade. (O meu conhecimento de TI é de usuário apenas.).

Assim, concluo a parte do projeto de estruturação na nuvem. Mas não acabou, é agora que a brincadeira vai começar de verdade. kk

4. **Quais arquivos serão analisados?**

Como falei anteriormente o Comex Stat é um sistema gratuito para consultas e extração de dados do comércio exterior brasileiro. No site (comexstat.mdic.gov.br) a base de dados do sistema está disponível para download, assim como as Tabelas Auxiliares. Apresenta dados desde 1997 até o último mês completo do ano atual.

O site recomenda : “**OS ARQUIVOS NÃO DEVEM SER UTILIZADOS EM SOFTWARES DE PLANILHA, CORRENDO O RISCO DE PERDA DE LINHAS E INFORMAÇÕES”.** Por serem volumes grandes de informações.

Utilizei a Tabela “**IMPORTAÇÃO 2020**”, que são dados de Janeiro a Julho deste ano.E ainda as Tabelas Auxiliares: VIA, PAÍSES, NCM e SH SISTEMA HARMONIZADO.

Vou explicar rapidamente o que significam NCM e SH, para ajudar no entendimento de mais adiante.

- SH - Sistema Harmonizado é um método internacional de classificação de mercadorias, baseado numa estrutura de código de 6 dígitos, com suas respectivas descrições.
- NCM - Nomenclatura Comum do Mercosul é baseado no SH, porém composto com 8 dígitos, serve para identificar a natureza dos produtos comercializados no Brasil e nos outros países do Mercosul, com o objetivo básico de **fazer o controle de mercadorias** e para **cálculo dos tributos.**

A estrutura do código NCM e SH é a mesma e utilizado em todo o mundo. Ele é dividido em Capítulo, Posição, Sub-capítulo, que são os seis primeiros dígitos e representam a classificação SH (Sistema Harmonizado) e os dois últimos dígitos são parte das especificações do próprio Mercosul.

Exemplo de NCM:

01. - Animais Vivos - 2 primeiros dígitos do SH
01.02. - Animais Vivos da espécie bovina - 4 primeiros dígitos do SH
01.02.29. - Bovinos domésticos - 6 primeiros dígitos do SH
01.02.29.1. - Para reprodução - 7° dígito do NCM
01.02.29.1.0 - Outros - 8° dígito do NCM

5. PgAdmin 4 - “Importar arquivo”

Anteriormente expliquei quais arquivos utilizei e de onde os copiei. Agora é preciso importá-los para o banco de dados para poder analisá-los.
No PgAdmin 4, conectado ao server, primeiro deve-se criar um Table (pré-arquivo) com o mesmo número de colunas do arquivo a ser estudado, especificando o tipo de cada coluna.

- Existe duas maneiras para fazer isso:
    - Utilizando o assistente do PgAdmin ou
    - criando manualmente com uma query

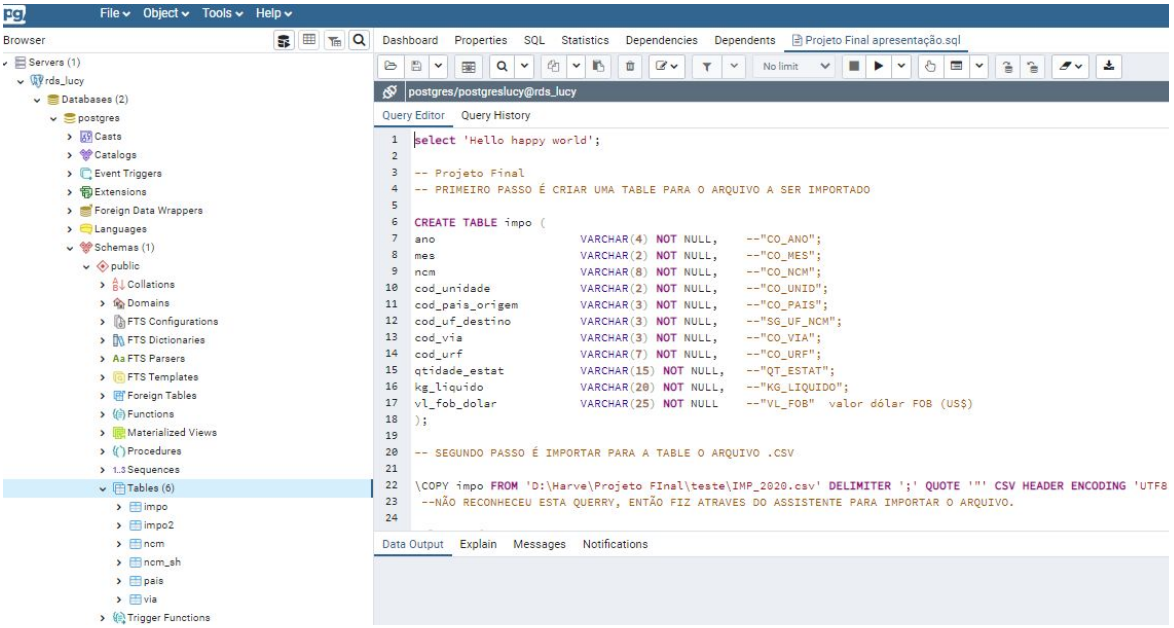
O passo seguinte é importar o arquivo para o Table criado. Simples, né!! Bem, nem tanto. Mas vamos lá.

Eu criei o table impo com a seguinte query:

```
CREATE TABLE impo (
ano                 VARCHAR(4) NOT NULL,    --"CO_ANO";
mes                 VARCHAR(2) NOT NULL,    --"CO_MES";
ncm                 VARCHAR(8) NOT NULL,    --"CO_NCM";
cod_unidade         VARCHAR(2) NOT NULL,    --"CO_UNID";
cod_pais_origem     VARCHAR(3) NOT NULL,    --"CO_PAIS";
cod_uf_destino      VARCHAR(3) NOT NULL,    --"SG_UF_NCM";
cod_via             VARCHAR(3) NOT NULL,    --"CO_VIA";
cod_urf             VARCHAR(7) NOT NULL,    --"CO_URF";
```

```
qtidade_estat           VARCHAR(15) NOT NULL,   --"QT_ESTAT";
kg_liquido              VARCHAR(20) NOT NULL,   --"KG_LIQUIDO";
vl_fob_dolar            VARCHAR(25) NOT NULL    --"VL_FOB"  valor FOB dólar   );
```

E importei o arquivo “impo.CSV” para o “table impo” utilizando o assistente, porque com a query que fiz, não foi possível. Novamente um erro de permissão! Mas com o assistente deu certo.



6. Análise dos dados das tabelas

Após concluída a etapa de importação da primeira tabela ao banco de dados. Vamos conhecê-la, como o professor falava, vamos namorar os dados:

```
select * from impo;
```
--Tabela tem 982.913 linhas e 11 colunas

```
select distinct ncm from impo;
```
--Tabela mostra 8.341 NCM´s diferentes (cada NCM representa um produto diferente)

```
select distinct cod_unidade from impo;
```
--13 unidades da RECEITA FEDERAL diferentes

```
select distinct cod_pais_origem from impo;
```
--Foram importados produtos de 226 países diferentes

```
select distinct cod_via from impo;
```
--10 vias de transporte diferentes

```
select distinct cod_uf_destino from impo;
```
--Tabela tem 28 estados = 26 estados + DF + ZN (Zona não declarada)
Zona não declarada? Me chamou atenção. Talvez um produto específico que não tem declarada o estado que está importando...estranho. Vamos ver mais de perto...

```
select * from impo where cod_uf_destino = 'ZN';
```

--Como resultado a query mostra 1.338 linhas, são NCM´s distintas que não constam o estado brasileiro importador, a princípio sem padrão específico. Precisaria uma análise mais profunda. Vamos adiante.

```
select cod_uf_destino, count(*) from impo group by cod_uf_destino order by count desc;
```
--Número de importações de cada estado organizado por ordem desc
--1° SP, 2° SC, 3° PR e 4°MG. O Paraná é o terceiro Estado que mais importou em quantidade de importações.

--Quais produtos o Paraná importou?
```
select ncm, count(*) from impo where cod_uf_destino = 'PR' group by ncm order by count desc;
```
--O Paraná importou 4.459 itens diferentes e as NCM´s mais utilizadas foram "39269090","40169300","73181500","40169990","73269090". Que produtos são esses?

--Quais meios de transporte foram utilizados para trazer ao Paraná?
```
select cod_via , count(*) from impo where cod_uf_destino = 'PR' group by cod_via order by count desc;
```
--7 tipos diferentes de via de transporte, os principais são 01, 04 e 07. O que significam?

--Gostaria de calcular o Valor FOB(*) por via de transporte.
(*) Para facilitar os processos de Importação e Exportação, foi criado pela Câmara de Comércio Internacional os ***Incoterms****. O Incoterm define a responsabilidade, os riscos e custos de cada parte, exportador e importador no transporte internacional.* ***FOB*** *(Free On Board) – Significa "Livre a bordo". Neste tipo de frete, o comprador assume todos os riscos e custos com o transporte da mercadoria, assim que ela é colocada a bordo do navio. É o Incoterm mais utilizado de um modo geral, por isso é utilizado nas estatísticas como referência. Nesta tabela estes valores estão sempre em dólar.*

```
select cod_via, sum(vl_fob_dolar) as total from impo group by cod_via;
--ERROR:  function sum(character varying) does not exist
--LINE 1: select cod_via, sum(vl_fob_dolar) as total from impo group b...
--HINT:  No function matches the given name and argument types. You might need to add explicit type casts.
```

--Esse erro rendeu mais uma mentoria, kkk. O erro ocorre porque quando criei a "Table", eu utilizei pra a coluna "vl_fob_dolar " o tipo varchar.
*--****Tipo varchar*** *= pode ser qualquer tipo de caractere, inclusive número, é genérico, mas* ***não é possível fazer cálculos*** *com ele.*
--Para resolver é possível utilizar o "::numeric", é a ***transformação temporária*** *dos valores da coluna em número.*
--Então utilizando a query anterior com o ::numeric.
```
select cod_via, sum(vl_fob_dolar::numeric) as total
from impo
group by cod_via
order by total desc;
```
--Aí sim funcionou, trouxe os valores que eu esperava. Mas, por ser uma mudança temporária, eu precisaria utilizar para cada query que eu fizesse. E aqui já percebi um problema. Eu já tinha que me preocupar em fazer a query corretamente,analisar o contexto

e ainda lembrar que quando tivesse cálculo teria que mudar. Não, isso não iria dar certo!! Aumentaria as chances de erro, não me pareceu nada prático!

Outra maneira de resolver seria deletar esta tabela e importá-la novamente com a tipagem correta para cálculo. E foi o que eu fiz. Não deletei a tabela impo, neste momento, para poder mostrar na apresentação o que havia acontecido. Eu criei uma nova tabela chamada "impo2" e nas colunas onde eu imaginei que seria necessário algum cálculo, utilizei o tipo numérico.

```
--Novo arquivo "impo2" com coluna numérica!!
CREATE TABLE impo2 (
ano                 NUMERIC(4) NOT NULL,     --"CO_ANO";
mes                 NUMERIC(2) NOT NULL,     --"CO_MES";
ncm                 VARCHAR(8) NOT NULL,     --"CO_NCM";
cod_unidade         VARCHAR(4) NOT NULL,     --"CO_UNID";
cod_pais_origem     VARCHAR(4) NOT NULL,     --"CO_PAIS";
cod_uf_destino      VARCHAR(3) NOT NULL,     --"SG_UF_NCM";
cod_via             VARCHAR(3) NOT NULL,     --"CO_VIA";
cod_urf             VARCHAR(8) NOT NULL,     --"CO_URF";
qtidade_estat       NUMERIC(20) NOT NULL,    --"QT_ESTAT";
kg_liquido          NUMERIC(20) NOT NULL,    --"KG_LIQUIDO";
vl_fob_dolar        NUMERIC(25) NOT NULL     --"VL_FOB" valor dólar FOB(US$)
);

select * from impo2;
--982.913 linhas, 11 colunas. A partir daqui utilizei, então apenas a tabela "impo2".

--Volto a query que tanto queria, o total de importações por valor e classificada por via de transporte. kkk
select cod_via, sum(vl_fob_dolar) as total_fob
from impo2
where cod_uf_destino = 'PR'
group by cod_via
order by total_fob desc;
 --E agora tranquilamente, aparece o resultado esperado. São sete vias de transporte diferentes utilizadas pelo Paraná.

select ncm, sum(vl_fob_dolar) as total_dolar
from impo2
where cod_uf_destino = 'PR'
group by ncm
order by total_dolar desc;
--O Paraná importou 4.459 itens diferentes, o mais vendido foi a NCM 2710.1921 = R$ 658.202.974, o que é isso?
```

--Para responder essa e outras perguntas ***é necessário importar*** *algumas* ***tabelas complementares****: NCM, VIA, PAÍSES. Elas que identificam o produto, a via de transporte e o país de origem, respectivamente.*

--Importar a tabela complementar NCM = códigos dos NCM´s com descrição.

```
CREATE TABLE ncm(
ncm                 VARCHAR(8) NOT NULL,
cod_unidade         VARCHAR(2) NOT NULL,
cod_sh6             VARCHAR(6) NOT NULL,
cod_ppe             VARCHAR(4) NOT NULL,
cod_ppi             VARCHAR(4) NOT NULL,
cod_fat_agreg       VARCHAR(3) NOT NULL,
cod_cuci_item       VARCHAR(4) NOT NULL,
cod_cgce_n3         VARCHAR(3) NOT NULL,
cod_siit            VARCHAR(4) NOT NULL,
cod_isic_classe     VARCHAR(4) NOT NULL,
cod_exp_subset      VARCHAR(4) NOT NULL,
ncm_portug          TEXT NOT NULL,
ncm_espanhol        TEXT NOT NULL,
ncm_ingles          TEXT NOT NULL,
 CONSTRAINT NCM_pkey PRIMARY KEY (ncm)
);
```

--O arquivo “NCM.csv” foi importado para a Table “ncm” através do assistente para importar arquivo.
--Porém deu erro ao importar a tabela, dizendo que havia erro no "UTF8" na linha 2.
*--O arquivo .csv não estava utilizando a **codificação** "UTF8" e sim a 'ANSI', por isso não estava reconhecendo.*
*--Solução: Abrir o arquivo “NCM.csv” em um editor de texto (Notepad ++) e **converter a codificação** do arquivo para "UTF8" e salvar. Resolvido o problema de codificação! Depois reiniciar a importação do arquivo.*

--Reiniciada a importação através do assistente, porém novo erro na tabela. Erro nas linhas 12.113 a 12.120.
--Abri o arquivo “NCM.csv” no editor de texto novamente. A tabela NCM é composta por 14 colunas, destas, três colunas são a descrição da NCM, cada uma, em um idioma, em português, espanhol e inglês. Relembrando NCM é o código do produto importado! Então cada código tem uma descrição detalhada do item. Em alguns itens a descrição é bem extensa, por isso utilizei o tipo da coluna = Texto, porque não há limite de extensão.

***Porém** no “NCM.csv” a linha 12.113, por exemplo, foi interrompida na parte da descrição e a continuação foi para a linha seguinte. E assim sucessivamente por outras linhas. Ou seja, a linha seguinte estava sem os objetos como na linha anterior, apenas com parte de uma descrição.*

Esse erro impossibilitaria a importação do arquivo “NCM.csv”. Como eu poderia solucionar isso? Bem, analisando esse arquivo e pensando nas informações dele, juntamente, com a minha experiência em Comércio Exterior, cheguei a conclusão de que eu poderia utilizar outro arquivo complementar - SH Sistema Harmonizado.

O meu objetivo era fazer análises macro de importação, então não precisaria do código completo. Para ficar mais claro vou voltar ao slide onde expliquei a composição do NCM.

01. - Animais Vivos - 2 primeiros dígitos do SH
01.02. - Animais Vivos da espécie bovina - 4 primeiros dígitos do SH
01.02.29. - Bovinos domésticos - 6 primeiros dígitos do SH
01.02.29.1. - Para reprodução - 7° dígito do NCM
01.02.29.1.0 - Outros - 8° dígito do NCM

*Por exemplo a descrição da **NCM 0102.2919** é **Animais vivos da espécie bovina - outros bovinos para reprodução.***
*Como a minha ideia para este projeto é a análise macro das informações, seria suficiente saber que o item é: **Animais vivos da espécie bovina**. Ou seja, os quatro primeiros dígitos seriam suficientes, o que resultaria da tabela do Sistema Harmonizado.*

Então ao invés do arquivo NCM importei o arquivo SH (Sistema Harmonizado) para facilitar as análises dos produtos. Primeiro criando a Table com a seguinte query e depois com o assistente importando os dados para a Table.

```
CREATE TABLE ncm_SH (
co_SH6                  VARCHAR(6) NOT NULL,    --"CO_SH6";
nome_SH6_por            TEXT NOT NULL,           --"NO_SH6_POR";
nome_SH6_esp            TEXT NOT NULL,           --"NO_SH6_ESP";
nome_SH6_ing            TEXT  NOT NULL,          --"NO_SH6_ING";
co_SH4                  VARCHAR(4) NOT NULL,     --"CO_SH4";
nome_SH4_por            TEXT  NOT NULL,          --"NO_SH4_POR";
nome_SH4_esp            TEXT  NOT NULL,          --"NO_SH4_ESP";
nome_SH4_ing            TEXT  NOT NULL,          --"NO_SH4_ING";
co_SH2                  VARCHAR(2) NOT NULL,     --"CO_SH2";
nome_SH2_por            TEXT  NOT NULL,          --"NO_SH2_POR";
nome_SH2_esp            TEXT  NOT NULL,          --"NO_SH2_ESP";
nome_SH2_ing            TEXT  NOT NULL,          --"NO_SH2_ING";
co_ncm_secrom           VARCHAR(2) NOT NULL,     --"CO_NCM_SECROM";
nome_sec_por            TEXT   NOT NULL,         --"NO_SEC_POR";
nome_sec_esp            TEXT   NOT NULL,         --"NO_SEC_ESP";
nome_sec_ing            TEXT   NOT NULL          --"NO_SEC_ING"
);
```

--Para utilizar a tabela "ncm_SH" havia um outro porém, mas fácil de ser resolvido. Eu precisaria de uma coluna conectora entre a tabela "impo2" e "ncm_SH".

Criei uma nova coluna (ncm_sh) na tabela "impo2", onde transformei o código ncm (de 8 dígitos) em quatro dígitos (sh4), utilizando o substring.

```
ALTER TABLE impo2 ADD COLUMN ncm_sh VARCHAR(6);
update impo2 set ncm_sh = substring(ncm from 0 for 5);
```
--A "conexão" entre as duas tabelas "impo2" e "ncm_SH" estava pronta!

--As outras tabelas complementares foram importadas da mesma forma, Table criada com query e utilizando o assistente para importar o arquivo para Table.

--Arquivo PAÌS : código com descrição de países

```
CREATE TABLE pais(
cod_pais        VARCHAR(3) NOT NULL,                    --"CO_PAIS"
pais_ison3      VARCHAR(3) NOT NULL,                    --"CO_PAIS_ISON3"
pais_isoa3      VARCHAR(3) NOT NULL,                    --"CO_PAIS_ISOA3"
```

```
nome_port      CHARACTER VARYING(100)NOT NULL,          --"NO_PAIS"
nome_ingles    CHARACTER VARYING(100)NOT NULL,          --"NO_PAIS_ING"
nome_esp       CHARACTER VARYING(100) NOT NULL,         --;"NO_PAIS_ESP"
CONSTRAINT PAIS_pkey PRIMARY KEY ( cod_pais )
);

--Arquivo VIA : código com descrição das vias de transporte
CREATE TABLE via (
cod_via                 VARCHAR (2)  NOT NULL,
nome_via                CHARACTER VARYING (30)NOT NULL,
CONSTRAINT VIA_pkey PRIMARY KEY (cod_via)
);

 --Agora utilizando o inner join é possível ver todas as importações da NCM 0102.2190 - Animais vivos - outros bovinos reprodutores de raça pura e identificar de quais países importamos este produto.
select * from impo2 as i inner join pais as p on (i.cod_pais_origem = p.cod_pais)
where i.ncm = '01022190';
--No total foram 5 importações = 4 dos Estados Unidos e 1 da Argentina. Os Estados importadores foram Minas Gerais, São Paulo e Rio Grande do Sul.

--Eu quis mostrar aqui produtos diferentes dos tradicionais, como milho, soja, petróleo. Pegando outro item NCM 0101 - Cavalos, asininos e muares, vivos (inclui reprodutores de raça pura ou não). Utilizando o inner join para saber de que países importamos esse produto, selecionando algumas colunas (com somatório).
select i.cod_pais_origem, p.nome_port,
        sum(i.vl_fob_dolar) as total_dolar,
        sum(i.qtidade_estat) as qtdade_estatistica,
        sum(i.kg_liquido) as peso
from impo2 as i inner join pais as p on (i.cod_pais_origem = p.cod_pais)
where i.ncm_sh = '0101'
group by i.cod_pais_origem, p.nome_port
order by total_dolar desc;
--Os principais países que importamos esse produto são Países Baixos, Alemanha, Estados Unidos, França e Bélgica.

--De quais países (com nome) o Paraná mais importou?
select i.cod_pais_origem, pais.nome_port, sum(i.vl_fob_dolar) as total_fob,
sum(i.kg_liquido) as qtidade
from impo2 as i inner join pais on (i.cod_pais_origem = pais.cod_pais)
where i.cod_uf_destino = 'PR'
group by i.cod_pais_origem, pais.nome_port
order by total_fob desc;
--1. China  US$1.262.297.854; 2.Estados Unidos US$1.014.747.716;
3. Argentina US$395.141.252; 4. Paraguai US$301.035.038 e
5. Alemanha US$274.017.502

--Quais as vias de transporte mais utilizadas pelo Paraná nas Importações com somatório do valor FOB.
select i.cod_via, v.nome_via, sum(i.vl_fob_dolar) as total_fob
```

```
from impo2 as i inner join via as v on (i.cod_via = v.cod_via)
where i.cod_uf_destino = 'PR'
group by i.cod_via, v.nome_via
order by total_fob desc;
```
--Código *01 via MARÍTIMA US$5.153.835.179 e Código 07 via RODOVIÁRIA US$525.303.468 são as mais utilizadas. Além da via AÉREA $413.167.612, MEIOS PRÓPRIOS $139.890; CONDUTO/REDE DE TRANSMISSÃO $62.228; ENTRADA/SAÍDA FICTA**(*)** $13.200; POSTAL $4.916. **(*)** È um regime especial de importação.*

--Que produtos são importados por via = MEIO PROPRIO (cod 09)? com quantidade, valor e descrição do produto, classificado por valor em dólar

```
select i.ncm_sh, nsh.nome_sh4_por, sum(i.kg_liquido) as qtdade_via09,
sum(i.vl_fob_dolar)as total_dolar_via09
from impo2 as i inner join ncm_sh as nsh on (i.ncm_sh = nsh.co_sh4)
where i.cod_via ='09'
group by i.ncm_sh, nsh.nome_sh4_por
order by total_dolar_via09 desc;
```

--São 63 linhas/produtos diferentes com o somatório da quantidade e valor em dólar. O produto mais importado por esta via é a NCM 8802 = helicóptero e aviões. Mas aparecem por esta via, outros produtos como torneira e bomba para líquidos. Uma possibilidade para esclarecer - seriam produtos que foram importados na bagagem de viajante.
Porém observo que os valores totais estão incorretos. Quando eu executo a query sem o inner join o somatório é um, quando acrescento o inner join é outro. Não consegui identificar o porquê.

--Quais os produtos(NCM) que o Paraná mais importou de Janeiro a Julho de 2020?

```
select ncm_sh, sum(vl_fob_dolar) as total_dolar, sum(kg_liquido) as quantidade
from impo2
where cod_uf_destino = 'PR'
group by ncm_sh
order by total_dolar desc;
```

--São 941 linhas/produtos diferentes que totalizam "valor_FOB"=U$6.065.153.220.

--Agora fazendo um inner join com tabela "impo2" e "ncm_sh" com valor dólar e quantidade de cada produto.

```
select i.ncm_sh, nsh.nome_sh4_por, sum(i.vl_fob_dolar) as total_dolar, sum(i.kg_liquido) as
quantidade
from impo2 as i inner join ncm_sh as nsh on (i.ncm_sh = nsh.co_sh4)
where i.cod_uf_destino = 'PR'
group by i.ncm_sh, nsh.nome_sh4_por
order by sum(i.vl_fob_dolar) desc;
```

--Resulta em 941 linhas/produtos diferentes MAS que totalizam "valor_FOB"=U$56.043.985.495 e "Quantidade" 58.431.440.897. De novo, em algumas ocasiões, quando utilizo o inner join, o cálculo do somatório é feito corretamente, e em outras não. Infelizmente até o momento da apresentação não consegui identificar o porquê do erro. Visualizando apenas a descrição dos produtos, podemos ver os principais produtos importados pelo Paraná são: NCM 2720 = óleos de petróleo; 3808 = Inseticidas, rodenticidas, fungicidas, herbicidas; 3105 = Adubos (fertilizantes) minerais ou químicos; 8708= Partes e acessório de veículos. Demonstrando as características do Paraná, na area agrícola e automotiva.

--Quais produtos o Paraná importou em 2020 detalhado por mês?
Para não ficar muito grande a tabela, utilizei apenas os dois primeiros dígitos do SH. Utilizando o inner join com tabela "impo2" e "ncm_sh" (sistema harmonizado) acrescentando quantidade e sh2(dois primeiros dígitos) e classificado por mês.

```
select  nsh.co_sh2, i.ncm_sh, nsh.nome_sh4_por, i.mes, sum(i.kg_liquido) as quantidade,
sum(i.vl_fob_dolar) as total_dolar
from impo2 as i inner join ncm_sh as nsh on (i.ncm_sh = nsh.co_sh4)
where i.cod_uf_destino = 'PR'
group by nsh.co_sh2, i.ncm_sh,  nsh.nome_sh4_por, i.mes
order by i.mes desc;  -
```

--Resultaram 5037 linhas.
--Resolvi tentar copiar/exportar o resultado da query para poder utilizar como planilha e utilizar de outra forma.

```
copy(
select  nsh.co_sh2, i.ncm_sh, nsh.nome_sh4_por, i.mes, sum(i.kg_liquido) as quantidade,
sum(i.vl_fob_dolar) as total_dolar
from impo2 as i inner join ncm_sh as nsh on (i.ncm_sh = nsh.co_sh4)
where i.cod_uf_destino = 'PR'
group by nsh.co_sh2, i.ncm_sh,  nsh.nome_sh4_por, i.mes
order by i.mes desc)
to 'C:\Users\luciana\Documentos\resultado_teste.csv' delimiter ';' QUOTE '"'  csv header;
```

--ERROR: must be superuser or a member of the pg_write_server_files role to COPY to a file --
HINT: Anyone can COPY to stdout or from stdin. psql's \copy command also works for anyone.
--Não foi possível por não ter permissão para exportar ou copiar.

--Então tentei com uma tabela menor, sem inner join. Eu quis trazer itens diferentes para a apresentação, não apenas os tradicionais, como soja, petróleo ou milho. Como estamos vivendo uma pandemia peguei os códigos classificados pela categoria de produtos farmacêuticos (sh2 = 30) importados pelo Paraná, mostrando a quantidade e valor em dólar, classificados mês a mês.

```
select mes, ncm_sh, sum(kg_liquido) as qtidade, sum(vl_fob_dolar) as total_dolar
from impo2
where cod_uf_destino = 'PR' and  ncm_sh like '30%'
group by mes, ncm_sh
order by mes;
```

*--Resultou 42 linhas. Tentei fazer a cópia e novamente deu o erro de permissão. Então resolvi fazer manualmente. Ctrl C e Ctrl V funcionou bem!!kkk Dá mais trabalho, mas resolvi. Copiei para uma planilha do Excel e transformei no arquivo produtos_farmaceutico.**csv.***

--A segunda tabela que eu exportei manualmente foi a que eu já mostrei anteriormente:
--De quais países (com nome) o Paraná mais importou?

```
select i.cod_pais_origem, pais.nome_port, sum(i.vl_fob_dolar) as total_fob,
sum(i.kg_liquido) as qtidade
from impo2 as i inner join pais on (i.cod_pais_origem = pais.cod_pais)
where i.cod_uf_destino = 'PR'
group by i.cod_pais_origem, pais.nome_port
order by total_fob desc;
```

*--Resultou 121 linhas ou países. Aqui o cálculo dos valores totais está correto, mesmo com o inner join. O resultado desta query eu copiei para o arquivo paises_pr**.csv.***

7. Visualização com Power BI

--Utilizei o arquivo produtos_farmaceutico.csv no Power BI para fazer dois gráficos.
--No Power BI montei a tabela para visualizar melhor as informações. Colunas de 1 a 7 são os meses de Janeiro a Julho, com as quantidades totais dos produtos, da categoria.

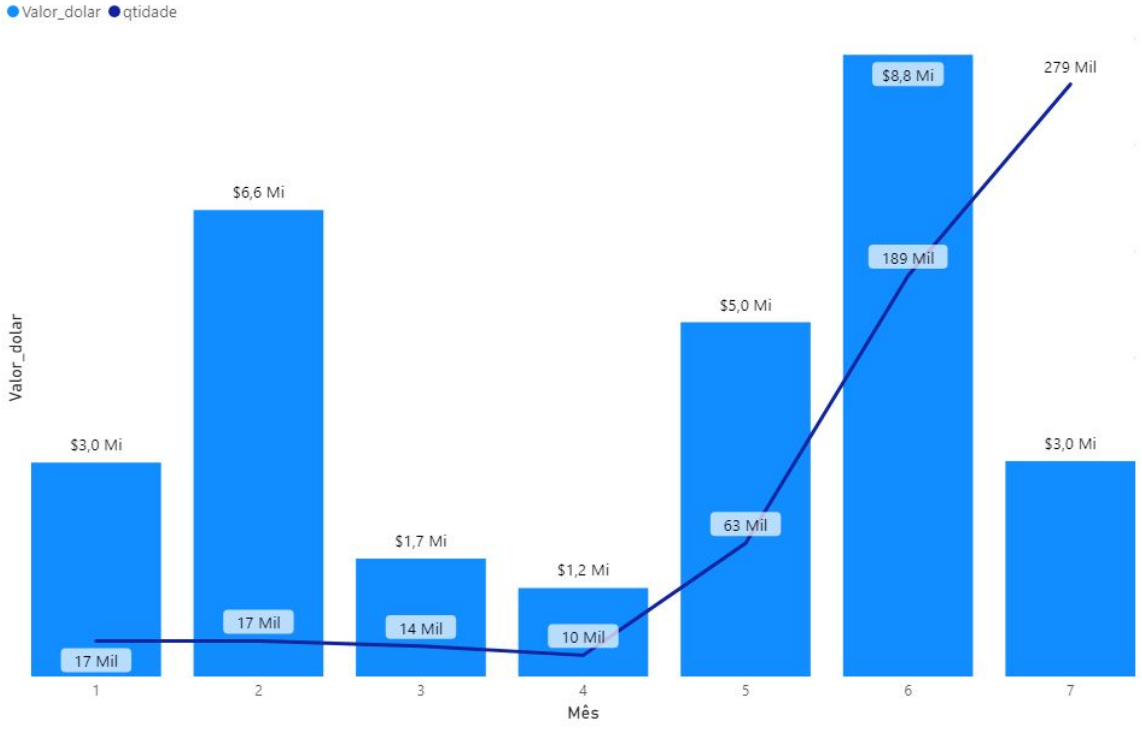
sh4	1	2	3	4	5	6	7	Total
3001	207	193	307	529	102	57	120	**1515**
3002	16668	16747	14218	9970	62676	188542	278671	**587492**
3003	16367	582	16770	50	22	17252	16138	**67181**
3004	115927	49596	98903	115400	63579	84605	102791	**630801**
3005	14772	1368	1169	36934	5291	5068	145	**64747**
3006	6383	3864	8131	4088	3722	3832	7020	**37040**
Total	**170324**	**72350**	**139498**	**166971**	**135392**	**299356**	**404885**	**1388776**

--E nesta de Janeiro a Julho com os valores totais em dólar.

sh4	1	2	3	4	5	6	7	Total
3001	$100.033	$107.894	$151.985	$199.374	$52.706	$23.531	$46.353	**$681.876**
3002	$3.019.161	$6.584.085	$1.663.246	$1.248.080	$5.000.711	$8.775.556	$3.040.899	**$29.331.738**
3003	$95.912	$17.074	$73.850	$11.250	$14.936	$115.784	$50.708	**$379.514**
3004	$21.296.162	$10.216.590	$16.236.705	$17.094.722	$6.123.332	$11.788.687	$15.185.301	**$97.941.499**
3005	$93.108	$14.423	$64.848	$303.398	$31.318	$30.095	$14.574	**$551.764**
3006	$465.640	$472.983	$485.057	$398.825	$265.575	$357.376	$326.425	**$2.771.881**
Total	**$25.070.016**	**$17.413.049**	**$18.675.691**	**$19.255.649**	**$11.488.578**	**$21.091.029**	**$18.664.260**	**$131.658.272**

--Das duas tabelas dois itens se sobressaem 3002 e 3004.

Importações do Estado do Paraná para NCM 3002



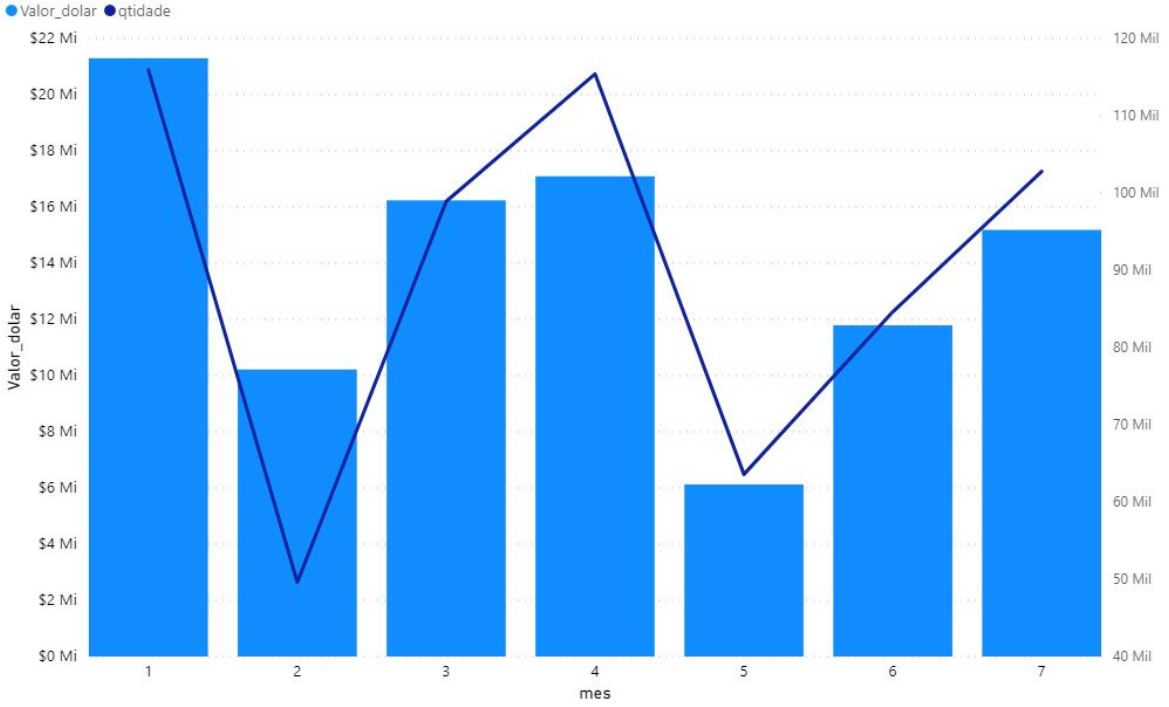
3002 - Sangue humano; sangue animal preparado para usos terapêuticos, profiláticos ou de diagnóstico; anti-soros, outras frações do sangue, **produtos imunológicos modificados,** mesmo obtidos por via biotecnológica; **vacinas,** toxinas, culturas de microrganismos.
(*)Inclui Kits de teste para Covid-19, baseados em reações imunológicas

--As colunas indicam os valores totais em dólar de cada mês. A linha azul escura indica a quantidade total do respectivo mês.
--É possível ver claramente o aumento da quantidade e do valor total a partir de Maio. Nesta NCM estão registrados os kits de teste para a Covid-19. A pandemia no Brasil iniciou em meados de Março, os testes devem ter sido comprados no final de Março ou início de Abril, chegando no país a partir de Maio. No Comércio nacional você compra hoje e recebe, em sua casa, em alguns dias, no Comércio Internacional não funciona assim. Normalmente leva de trinta a noventa dias, ou até mais tempo, dependendo se o fornecedor tem o produto em estoque, ou vai produzir após o pedido.

O aumento das quantidades são bem expressivas de 10 mil unidades em abril, vai para 63 mil em maio, triplicando em junho e chegando em 279 mil unidades em julho.
Em fevereiro a quantidade foi a mesma de janeiro, mas o valor duplicou. Seria necessário uma análise mais profunda, mas eu poderia imaginar que foi importado algum item de valor agregado alto. E em julho poderíamos pensar ao contrário. O preço de algum item baixou em comparação ao pago nos meses anteriores.

--O segundo item com valores expressivos foi o 3004, que apresento no gráfico a seguir.

Importações do Estado do Paraná para NCM 3004



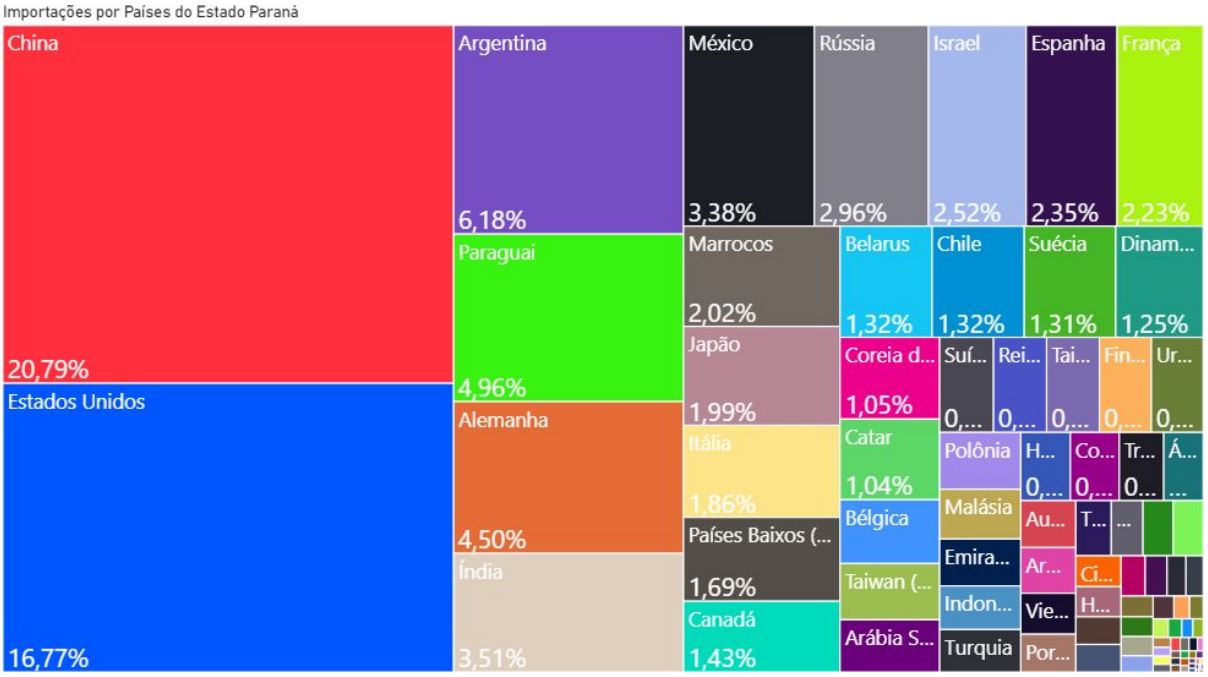
3004 - Medicamentos (exceto os produtos das posições 3002, 3005 ou 3006) constituídos por produtos misturados ou não misturados, preparados para fins terapêuticos ou profilácticos, apresentados em doses (incluindo os destinados a serem administrados por via subcutânea.

--As colunas indicam os valores totais (esquerda) em dólar de cada mês. A linha azul escura indica a quantidade total do respectivo mês. Nesse gráfico não é possível perceber

nenhuma tendência. A quantidade e o valor são equivalentes em todos os meses. O ideal seria analisar com outros anos para verificar se há alguma sazonalidade que explicaria os valores altos de janeiro ou a redução de fevereiro e maio, a cada dois meses.

*--O outro arquivo foi o paises_pr**.csv,** que mostra os valores totais e quantidades dos países que o Paraná mais importou de Janeiro a Julho de 2020.*

Importações do Estado do Paraná por países



--São os valores totais em porcentagem do total, de cada país que o Paraná importou.
--Ou seja do total de importações do Paraná, 20,79% foram da China, em seguida ficou os Estados Unidos com 16,77% e em terceiro lugar são as importações da Argentina com 6,18%. Cada cor representa um país diferente.

--Muito obrigada, esse foi o meu primeiro projeto de Ciência de Dados! Um desafio pessoal, que com muito orgulho apresentei hoje para vocês. Não foi um projeto fácil, exigiu muita pesquisa, leitura e estudo, mas me mostrou que é possível. Agradeço o suporte dos professores, que sempre muito atenciosos, me ajudaram muito, em especial o Professor Charles. **Muito obrigada!!**

Referências:

- http://comexstat.mdic.gov.br/pt/home
- http://www.mdic.gov.br/comercio-exterior/estatisticas-de-comercio-exterior/
- https://www.escolalinux.com.br/blog/o-que-e-aws-pra-que-serve-e-por-que-devo-domina-la
- https://www.opservices.com.br/principais-servicos-da-aws-amazon-web-services/
- https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_GettingStarted.CreatingConnecting.PostgreSQL.html
- https://www.postgresql.org/
- https://signin.aws.amazon.com/