1. Crie um banco local e as tabelas que iram receber os inserts contidos nos .sql;

Para a criação do banco eu utilizei do mySql o codigo usado em sql foi(Dumb do mesmo na pasta de arquivos):

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_D
ATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----------------------------------------------------
-- Schema mydb
-- -----------------------------------------------------

-- -----------------------------------------------------
-- Schema mydb
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
USE `mydb` ;

-- -----------------------------------------------------
-- Table `mydb`.`usuarios`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`usuarios` (
  `id` INT NOT NULL,
  `medico` TINYINT NOT NULL,
  `adm` TINYINT(99) NOT NULL,
  `especialidades` VARCHAR(255) NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) VISIBLE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `mydb`.`pacientes`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`pacientes` (
  `id` INT NOT NULL,
  `nascimento` DATETIME NULL,
  `pais` VARCHAR(10) NULL,
  `estadocivil` VARCHAR(45) NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) VISIBLE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `mydb`.`procedimentos`
```

```sql
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`procedimentos` (
  `id` INT NOT NULL,
  `tipo` VARCHAR(45) NULL,
  `descricao` VARCHAR(255) NULL,
  `tuss` VARCHAR(255) NULL,
  `deletado` TINYINT NOT NULL,
  `valor` DOUBLE NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) VISIBLE)
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `mydb`.`agendamentos`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`agendamentos` (
  `id` INT NOT NULL,
  `idPcte` INT NOT NULL,
  `data` DATETIME NOT NULL,
  `convenio` VARCHAR(45) NULL,
  `tipo` VARCHAR(45) NOT NULL,
  `hora` TIME NULL,
  `medico` INT NOT NULL,
  `status` TINYINT(98) NOT NULL,
  `valor` DOUBLE NULL,
  `idProcedimento` INT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) VISIBLE,
  INDEX `fk_agendamentos_pacientes_idx` (`idPcte` ASC) VISIBLE,
  INDEX `fk_agendamentos_usuarios1_idx` (`medico` ASC) VISIBLE,
  INDEX `fk_agendamentos_procedimentos1_idx` (`idProcedimento` ASC)
VISIBLE,
  CONSTRAINT `fk_agendamentos_pacientes`
    FOREIGN KEY (`idPcte`)
    REFERENCES `mydb`.`pacientes` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_agendamentos_usuarios1`
    FOREIGN KEY (`medico`)
    REFERENCES `mydb`.`usuarios` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_agendamentos_procedimentos1`
    FOREIGN KEY (`idProcedimento`)
    REFERENCES `mydb`.`procedimentos` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `mydb`.`agenda_status`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`agenda_status` (
  `id` INT NOT NULL,
  `status` VARCHAR(45) NOT NULL,
  `cor` VARCHAR(7) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) VISIBLE)
ENGINE = InnoDB;


SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Obs: alguns dos dados estvam errados, porem filtrei e ajeitei os mesmo(que estão disponíveis na pasta arquivos)

Comentários tabela usuarios:

Alguns ids estavam “NULL”, como não havia nenhum tipo de informação relevante e é boa pratica deixar o ID sem a possibilidade de ser "NULL” e não ter reedito, decidi deletar esses dados.

o id 10 esta repetido então modifiquei o segundo que aparecia para 1110

o id 877 estava repetido então modifiquei para 1111 do segundo que aparecia

o id 1 estava duplicado então modifiquei para 1112

medico - Como não havia nenhum dado “NULL” botei como “ not null ”

adm - Como o maior valor dessa coluna foi 99 botei o limite como 99 e como não havia nenhum “NULL” deixei como “not null”

Comentários tabela pacientes:

Havia alguns dados que estavam todos “NULL” exceto o id, por isso deixei que todas as colunas exceto o id podendo aceitar o “NULL”

comentários tabela procedimentos:

Nessa tabela tirando o ID a única coluna que não possuía nenhum “NULL” era “deletado” por isso botei ela como “not null”

(alguns dados possuíam tudo NULL (tirando “id” e “deletado”), estavam com deletado = 1, porém não sabia se era para deletar esses dados vistos que ainda possuíam id)

comentários tabela agendamentos:

Como “IdPcte” é necessário para fazer um agendamento, pois é necessário um cliente para fazer agendamento coloquei como “not null”

Como não havia nenhum “NULL” nas colunas “data”,” tipo”,” status “e “medico” eu as botei como “not null’’

na coluna hora existia alguns dados que tinham como hora “-01:00:00” modifiquei esses casos para “01:00:00”

em status havia alguns -1,-2 porém não sei se isso é um status valido por isso deixei assim.

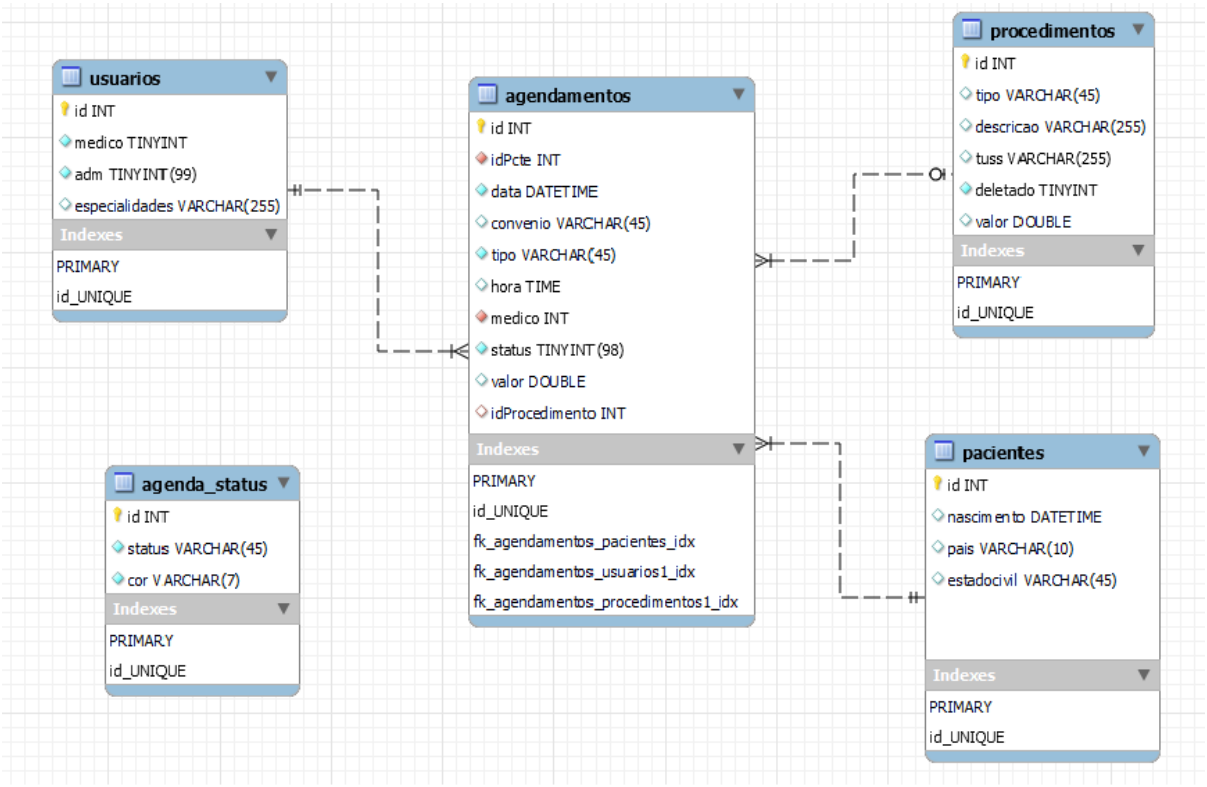
nessa tabela muitos das chaves estrangeiras não existiam em outras tabelas, então minha ideia inicial foi substituir os dados que estavam faltando por dados que existiam, porem ia comprometer a análise dos dados do mesmo, por conta disso tive que criar uma tabela sem ligações com as outras para poder fazer as consultas SQL

comentários tabela agenda_status:

havia algumas linhas com esse código “INSERT INTO `` (id,status,cor) VALUES (0,NULL,NULL);” como não havia nenhum dado útil e já existia um id com 0 apaguei esses dados, também havia algumas id com números negativos como não sabia se estava nas regras de negócio por isso não modifiquei deixei assim

como tirando os que excluir nenhuma das linhas tinha um “NULL” coloquei not null em todos

1.2 Crie uma pequena modelagem contendo as relações e cardinalidades das tabelas:



2. Consultas sql:

2.1. Top 10 médicos que mais realizaram agendamentos e quais os procedimentos mais realizados por eles;

```
SELECT medico, COUNT(*)
FROM agendamentos
Group By medico
Order By COUNT(*) DESC
LIMIT 10
```

Resultado:

medico	COUNT(*)
876	242
877	199
666	141
894	114
875	76
4	68
880	56
58	28
878	19
899	15

2.2. Em ordem crescente os meses de agendamentos e as quantidades de agendamentos

```
SELECT Month(data), COUNT(*)
FROM agendamentos
GROUP BY Month(data)
ORDER BY COUNT(*) DESC;
```

Resultado:

Month(data)	COUNT(*)
1	158
12	152
9	134
8	122
10	120
7	100
11	81
2	69
3	27
4	17
6	6
5	5

2.3. retorne paciente, data, hora, procedimento realizado, medico e status de todos os agendamentos;

```
SELECT
agendamentos.idPcte,pacientes.nascimento,pacientes.pais,pa
cientes.estadocivil,
```

```
agendamentos.data,
agendamentos.hora,agendamentos.idProcedimento,
procedimentos.tipo,procedimentos.descricao,procedimentos.t
uss,
procedimentos.deletado,procedimentos.valor,
agendamentos.medico,usuarios.especialidade,agendamentos.st
atus
FROM agendamentos
INNER JOIN pacientes ON agendamentos.idPcte = pacientes.id
INNER JOIN procedimentos ON agendamentos.idProcedimento =
procedimentos.id
INNER JOIN usuarios ON agendamentos.medico = usuarios.id;
```

Resultado:

Obs: Como havia falado nos comentários da tabela, não pude adicionar os dados na tabela por conta de erros com a chave estrangeira, por isso coloquei todos os dados com "1" para poder testar se a consulta estava funcionando

idPcte	nascimento	pais	estadocivil	data	hora	idProcedimento	tipo	descricao	tuss	deletado	valor	medico	especialidade	statuss
1	1	1	1	2018-07-24	00:00:01	1	1	1	1	1	1	1	1	1

3. (opcional) Realize uma análise dos dados do banco. Pode ser qualquer análise.

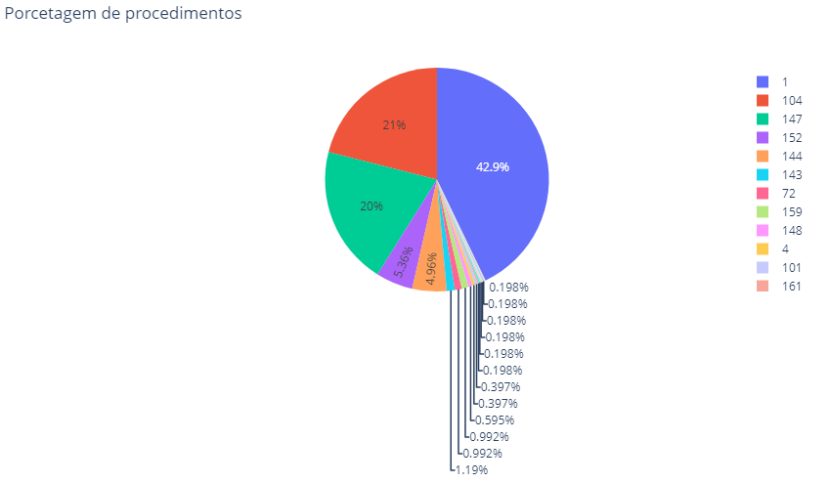
Para fazer os gráficos eu utilizei o pandas juntamente ao plotly na IDE jypter(o arquivo .ipynb esta na pasta arquivos)

1- Porcentagem de procedimentos realizados por Id

Código:

```
import pandas as pd
import plotly.express as px

tabelas = pd.read_excel("plan.xlsx")
tabelas_paciente = pd.read_excel("paciente.xlsx")

#Porcetagem de procedimentos
procedimentos = tabelas[['valor', 'idProcedimento']]
df = pd.DataFrame(procedimentos)
df = df.dropna()
fig = px.pie(df, names='idProcedimento',title='Porcetagem de
procedimentos')
fig.show()
```

Resultado:



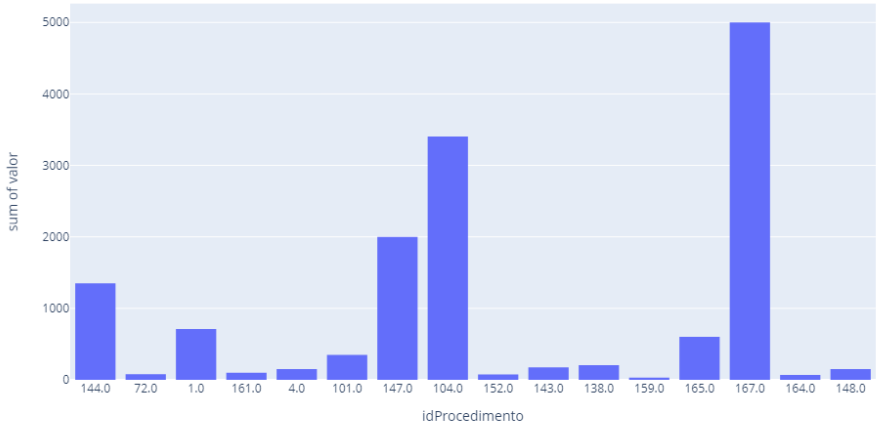
Analise dados:

Grande parte dos procedimentos são realizado menos de 1% das vezes, cabe uma investigação para saber se o preço deles estão muitos altos, se a poucos médicos se realizam ou se são procedimentos raros.

2- Procedimentos que mais geraram dinheiro

```
#cod 615 e o 481 estavam com um valor irreal que esta quebrando a analise de dados

df = df.drop(615)

df = df.drop(481)

df['idProcedimento'] = df['idProcedimento'].astype(str)

df = df.dropna()
df = df.drop_duplicates()
grafico = px.histogram(data_frame=df, x='idProcedimento', y='valor')
grafico.show()
```

Resultado:



Analise:

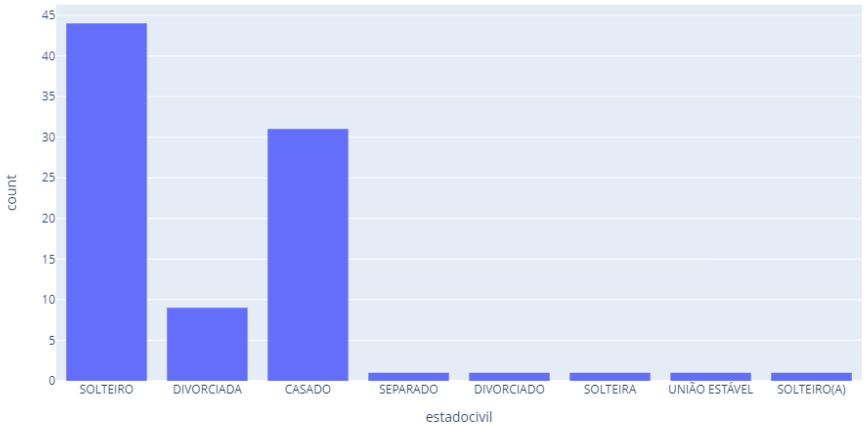
Alguns procedimentos não dinheiro quase nenhum, cabe uma investigação para ver se vale a pena manter a oferta dos mesmos

3- Grafico de estado civil

Codigo:

```
pacientes = tabelas_paciente['estadocivil']
df = pd.DataFrame(pacientes)
df = df.dropna()
grafico = px.histogram(data_frame=df, x='estadocivil')
grafico.show()
```

Resultado:



Analise:

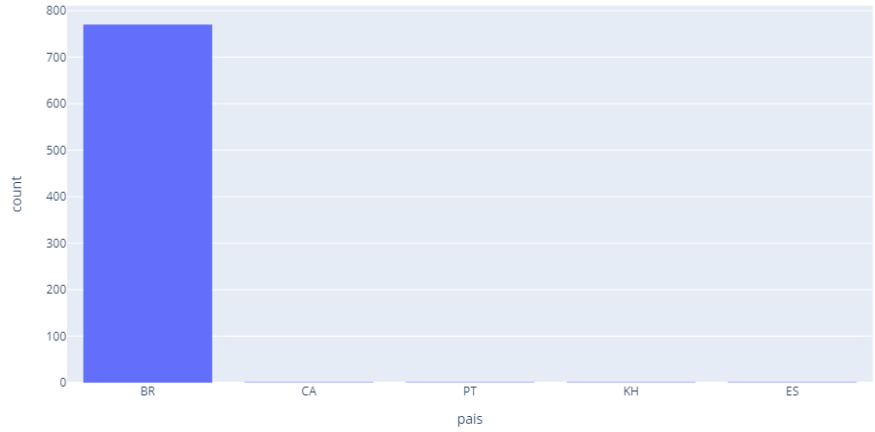
Muito dos dados dos clientes não tinha essa informação e não há padrão de escrita em alguns dados como solteiro e solteiro(a), indicar a quem está registrando os dados para prestar mais atenção nisso ou transformar em Enuns para que não acha erro. Pode ser feito um programa de desconto para casais para incentivar mais deles a entrar na plataforma.

4- países dos clientes

Codigo:

```
pacientes = tabelas_paciente['pais']
df = pd.DataFrame(pacientes)
df = df.dropna()
grafico = px.histogram(data_frame=df, x='pais')
grafico.show()
```

Resultado:



Analise:

Rever as estratégias de marketing direcionadas a outros países, pois a atual não está nem um pouco eficiente,visto que apenas 5 pacientes são de outros paises, talvez buscar médicos que falem outro idioma seja uma alternativa boa, além do suporte em outros idiomas também .