

```
!pip install pandas
!pip install seaborn
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (1.3
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/di
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (fr
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/
Requirement already satisfied: seaborn in /usr/local/lib/python3.7/dist-packages (0.1
Requirement already satisfied: pandas>=0.23 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: numpy>=1.15 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: matplotlib>=2.2 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: scipy>=1.0 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: pyparsing!=2.0.4,!2.1.2,!2.1.6,>=2.0.1 in /usr/local
Requirement already satisfied: cyclr>=0.10 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (fr
```

ANÁLISE E RECOMENDAÇÃO E SOBRE A MODELAGEM LÓGICA E FÍSICA DO BANCO DE DADOS

Durante a análise feita nos modelos relacional, modelo

- ▼ lógico e físico da base dados verificou-se violação de integridade referencial nas seguintes tabelas:

- 1 - Appointment
- 2 - Prescribes
- 3 - Undergoes

As tabelas em questão permitem que os atributos FK nelas contidas armazenem valores NULL.

Recomendações

Para solucionar a violação de integridade verificada acima, recomendamos que se faça contatos com os analistas que projetaram o sistema de BD e que se retire o valor NULLABLE nos respectivos atributos no modelo lógico e que se defina como INTEGER NOT NULL nas tabelas onde aparecem como INT apenas.

```
!pip install pandas
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (1.3.4)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-packages (1.21.0)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (2022.1)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (1.16.0)
```

```
import pandas as pd
import numpy as np
import seaborn as sns
from datetime import datetime
```

```
pd.set_option('display.max_rows', 1000)
pd.set_option('display.max_columns', 1000)
```

```
df_affiliated = pd.read_csv('Affiliated_With.csv', sep=';', names=['Physician', 'Department'])
```

```
#Dataset de médicos que são associados e não associados ao hospital e seus respectivos departamentos
df_affiliated
```

	Physician	Department	PrimaryAffiliation	
0	1	1	1	
1	2	1	1	

#Total de médicos associados

```
medicos_associados = df_affiliated[df_affiliated.PrimaryAffiliation == 1].count()
medicos_associados
```

```
Physician      9
Department     9
PrimaryAffiliation  9
dtype: int64
```

#Total de médicos não associados

```
medicos_associados = df_affiliated[(df_affiliated.PrimaryAffiliation == 0)].count()
medicos_associados
```

```
Physician      2
Department     2
PrimaryAffiliation  2
dtype: int64
```

O hospital possui 9 médicos associados e 2 terceirizados

#Dataset referente as consultas dos pacientes, data de inicio e fim das consultas, quais m

```
df_Appointment = pd.read_csv('Appointment.csv', sep=';', names=['AppointmentID', 'Patient'
df_Appointment
```

	AppointmentID	Patient	PrepNurse	Physician	StartTo	EndAt	Examinat
0	13216584	100000001	101.0	1	2008-04-24 10:00:00.000	2008-04-24 11:00:00.000	
1	26548913	100000002	101.0	2	2008-04-24 10:00:00.000	2008-04-24 11:00:00.000	
2	36549879	100000001	102.0	1	2008-04-25 10:00:00.000	2008-04-25 11:00:00.000	
3	46846589	100000004	103.0	4	2008-04-25 10:00:00.000	2008-04-25 11:00:00.000	
4	59871321	100000004	NaN	4	2008-04-26 10:00:00.000	2008-04-26 11:00:00.000	
5	69879231	100000003	103.0	2	2008-04-26 11:00:00.000	2008-04-26 12:00:00.000	
					2008-04-26	2008-04-26	

```
total_consultas = df_Appointment.AppointmentID.count()
total_consultas
```

9

```
#Pacientes que mais realizaram consultas e datas das respectivas consultas
paciente_mais_consultado = df_Appointment[['Patient', 'StartTo', 'EndAt']].mode()
paciente_mais_consultado.drop(2, axis=0, inplace=True)
paciente_mais_consultado
```

	Patient	StartTo	EndAt	
0	100000001.0	2008-04-24 10:00:00.000	2008-04-24 11:00:00.000	
1	100000004.0	2008-04-25 10:00:00.000	2008-04-25 11:00:00.000	

```
#Médico que mais realizou consulta
medicos_mais_consultaram = df_Appointment[['AppointmentID', 'Physician']].mode()
medicos_mais_consultaram.drop(range(1, 9), axis=0, inplace=True)
medicos_mais_consultaram
```

	AppointmentID	Physician	
0	13216584	2.0	

```
#Enfermeiro que mais atendeu paciente
enfermeiro_mais_atendimento = df_Appointment[['PrepNurse', 'StartTo', 'EndAt', 'ExaminationRoom']].mode()
enfermeiro_mais_atendimento.drop(range(1, 3), axis=0, inplace=True)
enfermeiro_mais_atendimento
```

	PrepNurse	StartTo	EndAt	ExaminationRoom	
0	101.0	2008-04-24 10:00:00.000	2008-04-24 11:00:00.000	A	

```
#Enfermeiras que agendaram consultas de pacientes e não se sabe quem foram
df_Appointment[df_Appointment.isnull().any(axis=1)]
```

	AppointmentID	Patient	PrepNurse	Physician	StartTo	EndAt	ExaminationRoom
4	59871321	100000004	NaN	4	2008-04-26 10:00:00.000	2008-04-26 11:00:00.000	
6	76983231	100000001	NaN	3	2008-04-26 12:00:00.000	2008-04-26 13:00:00.000	

Foram realizadas 9 consultas durante o mês 04/2008 nos horários das 10:00 até as 13:00.

Os pacientes que mais realizaram consultas nesse mesmo período foram os pacientes com os números de identificação 100000001 e 100000004.

O médico que mais realizou consulta nesse período é o que possui o ID 2.

A enfermeira que mais realizou agendamento de consulta é que possui o ID 101.

Existem 2 enfermeiras que realizaram agendamento de consultas dos pacientes, porém suas indentidades são desconhecidas.

A tabela de Appoint apresenta problema de segurança de integridade referencial uma vez que permite que seja inserido valores nulos vindo da de enfermeiro.

```
df_Block = pd.read_csv('Block.csv', sep=';', names=['Floor', 'BlockCode'])
df_Block
```

	Floor	BlockCode
0	1	1
1	1	2
2	1	3
3	2	1
4	2	2
5	2	3
6	3	1
7	3	2
8	3	3
9	4	1
10	4	2
11	4	3

```
#Total de pisos no hospital
qtd_total_pisos = df_Block['Floor'].unique()
qtd_total_pisos
```

```
array([1, 2, 3, 4])
```

```
#Total de blocos no hospital
qtd_total_blocos = df_Block['BlockCode'].unique()
qtd_total_blocos
```

```
array([1, 2, 3])
```

O hospital possui tem 3 blocos e 4 pisos

```
df_Department= pd.read_csv('Department.csv', sep=';', names=['DepartmentID', 'Name', 'Head
df_Department
```

	DepartmentID	Name	Head	
0	1	General Medicine	4	
1	2	Surgery	7	
2	3	Psychiatry	9	

```
df_Medication = pd.read_csv('Medication.csv', sep=';', names=['Code', 'Name', 'Brand', 'De
df_Medication
```

	Code	Name	Brand	Description	
0	1	Procrastin-X	X	NaN	
1	2	Thesisin	Foo Labs	NaN	
2	3	Awakin	Bar Laboratories	NaN	
3	4	Crescavitin	Baz Industries	NaN	
4	5	Melioraurin	Snafu Pharmaceuticals	NaN	

Na tabela de Medicação na coluna Description foi inserido valores N/A, isto é não foram inseridas as informações técnicas e de validade dos medicamentos.

Recomendação: Atendendo a importância dessas informações técnicas e da validade, recomendamos ao hospital a obrigar os fornecedores desses medicamentos fornecerem tais informações e os DBAs ou engenheiros de dados do hospital devem cuidar que essas informações estejam sempre inseridas na coluna em questão.

```
df_Nurse = pd.read_csv('Nurse.csv', sep=';', names=['EmployeeID', 'Name', 'Position', 'Reg
df_Nurse
```

	EmployeeID	Name	Position	Registered	SSN	
0	101	Carla Espinosa	Head Nurse	1	111111110	
1	102	Laverne Roberts	Nurse	1	222222220	
2	103	Paul Flowers	Nurse	0	333333330	

```
#Total de enfermeiros que trabalham no hospital
total_enfermeiros = df_Nurse['EmployeeID'].count()
total_enfermeiros
```

3

O hospital tem cadastrado no seu sistema 3 funcionário na função de médicos

```
#Tabela de enfermeiros que trablham na escala de plantão
df_On_Call = pd.read_csv('On_Call.csv', sep=';', names=['Nurse', 'BlocFloor', 'BlockCode',
```

df_On_Call

	Nurse	BlocFloor	BlockCode	OnCallStart	OnCallEnd
0	101	1	1	2008-04-11 11:00:00.000	2008-04-11 19:00:00.000
1	101	1	2	2008-04-11 11:00:00.000	2008-04-11 19:00:00.000
2	102	1	3	2008-04-11 11:00:00.000	2008-04-11 19:00:00.000
3	103	1	1	2008-04-11 19:00:00.000	2008-05-11 03:00:00.000
4	103	1	2	2008-04-11 19:00:00.000	2008-05-11 03:00:00.000
5	103	1	3	2008-04-11 19:00:00.000	2008-05-11 03:00:00.000

```
#Enfermeira com maior número de plantão
maior_qtd_plantao = df_On_Call[['Nurse']].mode()
maior_qtd_plantao
```

	Nurse
0	103

O enfermeiro com o ID 103 é a que mais realizou plantão durante 1 mês, em pisos e blocos diferentes. Enquanto as o enfermeiro com o ID 101 teve 2 dois plantões por em 1 dia em blocos e pisos diferentes e o enfermeiro com o ID 102 teve apenas 1 plantão em um único bloco e piso

Recomendação: Contratar mais enfermeiras e melhorar o programa de escalas do hospital, para evitar o desgaste mental e físico dos mesmos.

```
df_Patient = pd.read_csv('Patient.csv', sep=';', names=['SSN', 'Name', 'Address', 'Phone', 'InsuranceID', 'PCP'],
df_Patient
```

	SSN	Name	Address	Phone	InsuranceID	PCP
0	100000001	John Smith	42 Foobar Lane	555-0256	68476213	1
1	100000002	Grace Ritchie	37 Snafu Drive	555-0512	36546321	2
2	100000003	Random J. Patient	101 Omgbqq Street	555-1204	65465421	2
3	100000004	Dennis Doe	1100 Foobaz Avenue	555-2048	68421879	3

```
total_paciente = df_Patient['SSN'].count()
total_paciente
```

4

O hospital tem na sua base de dados um total de 4 pacientes cadastrado, durante o tempo de suas atividades

Qual seria o motivo para tão pouca aderência de pacientes no hospital?

```
df_Physician = pd.read_csv('Physician.csv', sep=';', names=['EmployeeID', 'Name', 'Position', 'SSN'])
df_Physician
```

	EmployeeID	Name	Position	SSN
0	1	John Dorian	Staff Internist	111111111
1	2	Elliot Reid	Attending Physician	222222222
2	3	Christopher Turk	Surgical Attending Physician	333333333
3	4	Percival Cox	Senior Attending Physician	444444444
4	5	Bob Kelso	Head Chief of Medicine	555555555
5	6	Todd Quinlan	Surgical Attending Physician	666666666
6	7	John Wen	Surgical Attending Physician	777777777
7	8	Keith Dudemeister	MD Resident	888888888
8	9	Molly Clock	Attending Psychiatrist	999999999

```
#Total de médicos que trabalham no hospital
total_medicos = df_Physician['EmployeeID'].count()
total_medicos
```

9

O hospital possui um total de 9 médicos, dentre os quais 7 são médicos associados ao hospital e 2 terceirizados

Pela quantidade de paciente que o hospital tem, talvez se justifique a quantidade de médicos.

```
#Tabela com informações das prescrições médicas, com suas respectivas datas, matriculas, etc.
df_Prescribes = pd.read_csv('Prescribes.csv', sep=';', names=['Physician', 'Patient', 'Medication', 'Date', 'Appointment', 'Dose'])
df_Prescribes
```

	Physician	Patient	Medication	Date	Appointment	Dose
0	1	100000001	1	2008-04-24 10:47:00.000	13216584.0	5
1	9	100000004	2	2008-04-27 10:53:00.000	86213939.0	10
2	9	100000004	2	2008-04-30 16:53:00.000	NaN	5

O paciente com o ID 100000004 é o que mais teve prescrição médica e o médico que mais fez prescrição é o médico de ID 9

A coluna Appointment na sua última linha não foi cadastrado o ID referente a consulta do paciente.

Recomendação: Não preencher a coluna Appointment valores em branco

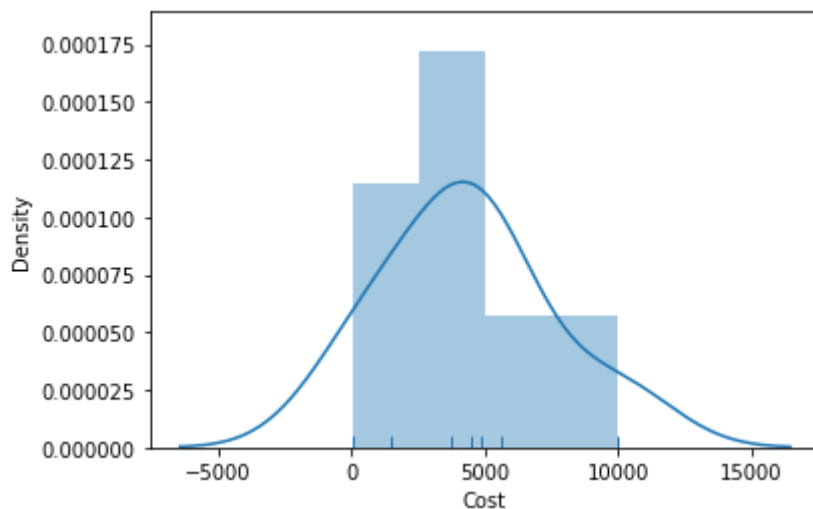
```
df_Procedures = pd.read_csv('Procedures.csv', sep=';', names=['Code', 'Name', 'Cost'])
df_Procedures
```

	Code	Name	Cost
0	1	Reverse Rhinopodoplasty	1500
1	2	Obtuse Pyloric Recombobulation	3750
2	3	Folded Demiophthalmectomy	4500
3	4	Complete Walletectomy	10000
4	5	Obfuscated Dermogastrotomy	4899
5	6	Reversible Pancreomyoplasty	5600
6	7	Follicular Demiectomy	25

#Distribuição dos valores de custo de procedimento

```
dist_valor_procedimento = sns.distplot(df_Procedures['Cost'], kde=True, rug=True, hist=True)
dist_valor_procedimento
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2103: FutureWarning:
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7fb9f1e46b90>
```



#Valor máximo do procedimento médico

```
maior_valor_procedimento_medico = df_Procedures[['Name', 'Cost']].max()
```

```
maior_valor_procedimento_medico
```

```
      Name    Reversible Pancreomyoplasty  
      Cost                                10000  
      dtype: object
```

```
#Valor minimo do procedimento médico
```

```
maior_valor_procedimento_medico = df_Procedures[['Name', 'Cost']].min()
```

```
maior_valor_procedimento_medico
```

```
      Name    Complete Walleectomy  
      Cost                                25  
      dtype: object
```

```
maior_valor_procedimento_medico = df_Procedures['Cost'].mean()
```

```
maior_valor_procedimento_medico
```

```
4324.857142857143
```

```
df_Room = pd.read_csv('Room.csv', sep=';', names=['RoomNumber', 'RoomType', 'BlockFloor'],  
df_Room
```

	RoomNumber	RoomType	BlockFloor	BlockCode	Unavailable	
0	101	Single	1	1	0	
1	102	Single	1	1	0	
2	103	Single	1	1	0	
3	111	Single	1	2	0	
4	112	Single	1	2	1	
5	113	Single	1	2	0	
6	121	Single	1	3	0	
7	122	Single	1	3	0	
8	123	Single	1	3	0	
9	201	Single	2	1	1	
10	202	Single	2	1	0	
11	203	Single	2	1	0	
12	211	Single	2	2	0	
13	212	Single	2	2	0	
14	213	Single	2	2	1	
15	221	Single	2	3	0	
16	222	Single	2	3	0	

```
#Quantidade de quartos no hospital
qtd_quartos = df_Room['RoomNumber'].count()
qtd_quartos
```

36

```
#Quantidade de salas por bloco
qtd_pisos_por_bloco = df_Room.groupby('BlockCode')[['RoomNumber']].count()
qtd_pisos_por_bloco
```


	RoomNumber	
BlockCode		
1	12	
2	12	
3	12	
4	12	
5	12	
6	12	
7	12	
8	12	
9	12	
10	12	
11	12	
12	12	
13	12	
14	12	
15	12	
16	12	

```
#Quantidade de quartos, ocupados e desocupados, agrupados por piso e por andar
```

21	412	Single	4	2	0
----	-----	--------	---	---	---

```
#Quantidade de quartos por piso
```

```
quartos_por_piso = df_Room.groupby('BlockFloor').agg({'RoomNumber': 'count'})
quartos_por_piso
```

	RoomNumber 
BlockFloor	
1	9
2	9
3	9
4	9

#Selecionar quartos ocupados e desocupados

#Desocupado

```
quartos_desocupados = df_Room[(df_Room.Unavailable == 0)].count()
quartos_desocupados
```

```
RoomNumber    29
RoomType       29
BlockFloor     29
BlockCode      29
Unavailable    29
dtype: int64
```

#Ocupado

```
quartos_ocupados = df_Room[(df_Room.Unavailable == 1)].count()
quartos_ocupados
```

```
RoomNumber     7
RoomType        7
BlockFloor      7
BlockCode       7
Unavailable     7
dtype: int64
```

O hospital tem uma lotação máxima de 36 quartos.

Está dividido em 3 blocos e

Recomendação: O hospital precisa de adotar políticas para facilitar que mais pacientes tenham acesso aos serviços do hospital, uma vez que há bastante salas sem serem usadas.

Seguro médico conveniado com o SUS

Liberar políticas para convenio com outros planos de saúde e assim atrair mais pacientes

Procurar mais profissionais para dar respostas as demandas que forem surgindo

#tabela de internação

```
df_Stay = pd.read_csv('Stay.csv', sep=';', names=['StayID', 'Patient', 'Room', 'StayStart', 'StayEnd'])
```

	StayID	Patient	Room	StayStart	StayEnd
0	3215	100000001	111	2008-01-05 00:00:00.000	2008-04-05 00:00:00.000
1	3216	100000003	123	2008-05-03 00:00:00.000	2008-05-14 00:00:00.000
2	3217	100000004	112	2008-02-05 00:00:00.000	2008-03-05 00:00:00.000

```
#Converter a coluna StayStart de object para datetime
df_Stay['StayStart'] = pd.to_datetime(df_Stay['StayStart'], format='%Y-%m-%d %H:%M:%S')
```

```
#Converter a coluna StayEnd de object para datetime
df_Stay['StayEnd'] = pd.to_datetime(df_Stay['StayEnd'], format='%Y-%m-%d %H:%M:%S')
```

```
#Criando a coluna tempo de internação do cliente e adicionando a mesma coluna no df
df_Stay['Tempo Internação'] = df_Stay['StayEnd'] - df_Stay['StayStart']
```

```
df_Stay.dtypes
```


```
StayID          int64
Patient         int64
Room            int64
StayStart       datetime64[ns]
StayEnd         datetime64[ns]
Tempo Internação timedelta64[ns]
dtype: object
```

```
#Pacientes com mais tempo de internação
df_Stay
```

	StayID	Patient	Room	StayStart	StayEnd	Tempo Internação
0	3215	100000001	111	2008-01-05	2008-04-05	91 days
1	3216	100000003	123	2008-05-03	2008-05-14	11 days
2	3217	100000004	112	2008-02-05	2008-03-05	29 days

O paciente 100000001 ficou mais tempo internado no hospital

```
#Tabela de certificação
df_Trained_In = pd.read_csv('Trained_In.csv', sep=';', names=['Physician', 'Treatment', 'C
df_Trained_In
```



	Physician	Treatment	CertificationDate	CertificationExpires
0	3	1	2008-01-01 00:00:00.000	2008-12-31 00:00:00.000
1	3	2	2008-01-01 00:00:00.000	2008-12-31 00:00:00.000
2	3	5	2008-01-01 00:00:00.000	2008-12-31 00:00:00.000
3	3	6	2008-01-01 00:00:00.000	2008-12-31 00:00:00.000
4	3	7	2008-01-01 00:00:00.000	2008-12-31 00:00:00.000
5	6	2	2008-01-01 00:00:00.000	2008-12-31 00:00:00.000
6	6	5	2007-01-01 00:00:00.000	2007-12-31 00:00:00.000
7	6	6	2008-01-01 00:00:00.000	2008-12-31 00:00:00.000
8	7	1	2008-01-01 00:00:00.000	2008-12-31 00:00:00.000
9	7	2	2008-01-01 00:00:00.000	2008-12-31 00:00:00.000
10	7	3	2008-01-01 00:00:00.000	2008-12-31 00:00:00.000
11	7	4	2008-01-01 00:00:00.000	2008-12-31 00:00:00.000
12	7	5	2008-01-01 00:00:00.000	2008-12-31 00:00:00.000

Clique duas vezes (ou pressione "Enter") para editar

```
#Tabela de pacientes que submeteram-se a algum tipo de procedimento médico
df_Undergoes = pd.read_csv('Undergoes.csv', sep=';', names=['Patient', 'Procedures', 'Stay
df_Undergoes
```

	Patient	Procedures	Stay	DateUndergoes	Physician	AssistingNurse
0	100000001	2	3215	2008-03-05 00:00:00.000	7	101.0
1	100000001	6	3215	2008-02-05 00:00:00.000	3	101.0
2	100000001	7	3217	2008-10-05 00:00:00.000	7	101.0
3	100000004	1	3217	2008-07-05 00:00:00.000	3	102.0
4	100000004	5	3217	2008-09-05 00:00:00.000	6	NaN

Tabela não pode permitir inserir valor nutable

O paciente 100000001 é quem ficou mais tempo internado no hospital total 91 dias e foi atendido por 2 médicos.

✓

0s

conclusão: 20:19

×