

Transformação do Dataset: Complaints of New York City

Mestrado em Ciência de Dados

UC: Gestão de Big Data

Docente: Pedro Ramos

1ºSemestre 2021/2022

Realizado por:

Afonso Fidanza, nº

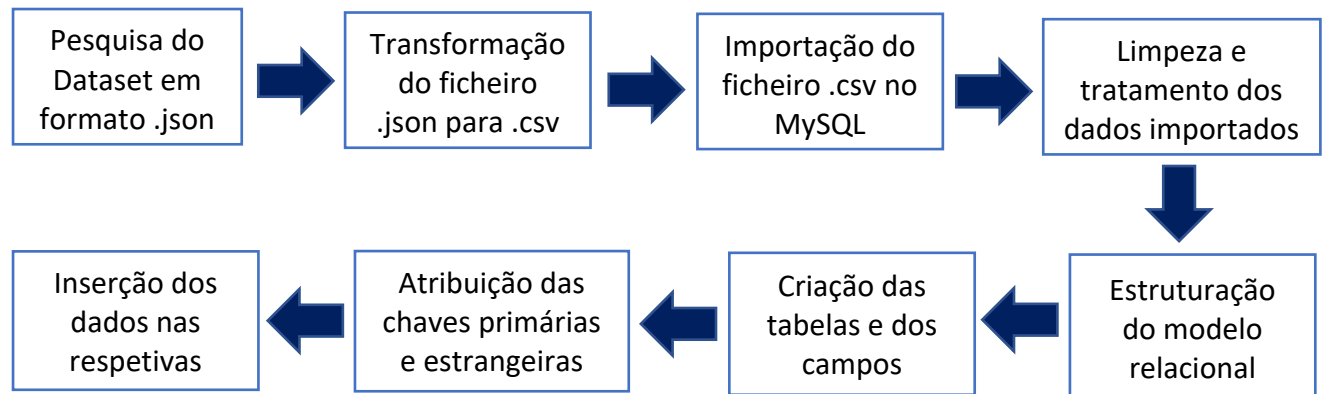
Bernardo Augusto, nº102033

Diana Pernas, nº87293

Índice

Introdução.....	3
Cronograma	Error! Bookmark not defined.
Sobre o Dataset.....	3
Transformação e limpeza do Dataset.....	4
De “.json” para “.csv”	4
Importação do .csv para o MySQL.....	4
Modelo Relacional.....	6
Esquema da Base de dados	6
Tabela Complain	7
Tabela Type	14
Tabela Unit	8
Tabela Space.....	8
Tabela Major	10

Cronograma



Sobre o Dataset

Este dataset contém dados da cidade de Nova York. A cidade tem uma plataforma de dados open source que é atualizada de acordo com a quantidade de dados que recebem.

Transformação e limpeza do Dataset

De “.json” para “.csv”

Para efetuar esta etapa da transformação do dataset que se encontra no formato de “.json” para “.csv”, foi necessário utilizar o MongoDB.

Para a importação do nosso ficheiro “.json” que contém o dataset foi necessário utilizar o comando visível no bloco 1.

```
mongoimport --db complain_problems --collection complains --drop
--file PATH/TO/FILE/complains.json
```

Bloco 1 - comando mongoimport para importar o dataset

Após importarmos o dataset para o MongoDB, verificou-se se os dados estavam importados corretamente e quais as “colunas” que fariam sentido para o nosso futuro modelo relacional. Para esta verificação foi utilizado o comando que é visível no bloco 2.

```
db.complains.find()
```

Bloco 2 - comando para verificação dos dados

Com este comando obtemos o output visível na figura 1.

```
{
  _id: ObjectId("61855059f2386d3375099ab8"),
  ProblemID: 14608374,
  ComplaintID: 6992771,
  UnitTypeID: 91,
  UnitType: 'APARTMENT',
  SpaceTypeID: 546,
  SpaceType: 'KITCHEN',
 TypeID: 3,
  Type: 'NON EMERGENCY',
  MajorCategoryID: 11,
  MajorCategory: 'GENERAL',
  MinorCategoryID: 73,
  MinorCategory: 'CABINET',
  CodeID: 679,
  Code: 'DAMAGED OR MISSING',
  StatusID: 2,
  Status: 'CLOSE',
  StatusDate: '2014-08-20T00:00:00.000',
  StatusDescription: 'The Department of Housing Preservation and Development in
pected the following conditions. Violations were issued. Information about speci
fic violations is available at www.nyc.gov/hpd.'
}
```

Figura 1 - output do comando db.complain.find()

Após uma análise cuidadosa foi decidido que seriam descartadas 2 “colunas”, “ProblemID” e “_id”.

O “ProblemID” foi excluído pois já podemos usar o “ComplainID” como chave primária da nossa futura tabela “Complain”.

O “_id” como é o ObjectID gerado automaticamente pelo mongo não tem qualquer valor para o nosso modelo.

Com os nossos campos já definidos pode-se agora avançar para a próxima fase que é a exportação dos dados em “.csv”. Para isto foi utilizado o comando que se encontra no bloco 3.

```
mongoexport --host localhost --db complain_problems --collection complains --
type=csv --out complains.csv --fields
ComplainID,UnitTypeID,UnitType,SpaceTypeID,SpaceType,TypeID,Type,MajorCategor
yID,MajorCategory,MinorCategoryID,MinorCategory,CodeID,Code,StatusID,Status,S
tatusDate,StatusDescription
```

Bloco 3 - comando para a exportação do dataset em “.csv”

Após executar o comando acima obteve-se o seguinte output, figura 2. Podendo verificar-se que foram exportados 3.449.697 registos.

```
2021-11-06T10:41:26.079+0000 [#####] complain_problems.co
mplains 3449697/3449697 (100.0%)
2021-11-06T10:41:26.079+0000 exported 3449697 records
```

Figura 2 - Output mongoexport

Importação do .csv para o MySQL

Na etapa anterior foi possível exportar o dataset no formato “.csv”. Nesta etapa o objetivo será a importação desse mesmo ficheiro para o MySQL.

Para a importação do nosso dataset para o MySQL foi necessário primeiro que tudo a criação de uma base de dados. Para isto foi utilizado o comando visível no bloco 4.

```
CREATE DATABASE complain_problems;
```

Bloco 4 - Comando para a criação da base de dados

Em seguida, situando-nos na base de dados criada foi necessário a criação de uma tabela “complain” com os seguintes campos: “ComplaintID”, “UnitTypeID”, “UnitType”, “SpaceTypeID”, “SpaceType”, “TypeID”, “Type”, “MajorCategoryID”, “MajorCategory”, “MinorCategoryID”, “MinorCategory”, “CodeID”, “Code”, “StatusID”, “Status”, “StatusDate” e “StatusDescription” para se poder depositar os dados. O comando para a criação desta tabela encontra-se no bloco 5.

```
CREATE TABLE Complain(ComplaintID INT, UnitTypeID INT, UnitType VARCHAR(100),
SpaceTypeID INT, SpaceType VARCHAR(100), TypeID INT, Type VARCHAR(100),
MajorCategoryID INT, MajorCategory VARCHAR(100), MinorCategoryID INT,
MinorCategory VARCHAR(100), CodeID INT, Code VARCHAR(100), StatusID INT, Status
VARCHAR(100), StatusDate DATETIME, StatusDescription TEXT);
```

Bloco 5 - Comando criação da tabela Complain

Após os passos da criação da base de dados e da tabela avançamos para a importação do dataset. Para a importação do dataset realizamos o comando apresentado no bloco 6.

```
LOAD DATA INFILE 'complains.csv'
INTO TABLE complain
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

Bloco 6 - Comando importação do dataset

Quando a importação terminou, verificou-se que a coluna “ComplainID” só continha zeros, figura 2.

ComplainID	UnitTypeID	UnitType	SpaceTypeID	SpaceType	TypeID	Type	MajorCategoryID	MajorCategory	MinorCategoryID	MinorCategory	CodeID	Code	StatusID	Status	StatusDate	StatusDescription
0	20	APARTMENT	68	ENTIRE APARTMENT	1	EMERGENCY	59	HEATHOT WATER	349	ENTIRE BUILDING	2717	NO HOT WATER	2	CLOSE	2014-08-22 00:00:00	The Department of Housing Preservation and Develop...
0	91	APARTMENT	545	ENTRANCE/FOYER	1	EMERGENCY	56	DOOR/WINDOW	333	DOOR	2665	LOCK BROKEN OR MISSING	2	CLOSE	2017-03-03 00:00:00	The Department of Housing Preservation and Develop...
0	91	APARTMENT	543	ENTIRE APARTMENT	1	EMERGENCY	56	DOOR/WINDOW	337	WINDOW FRAME	2636	LOOSE OR DEFECTIVE	2	CLOSE	2017-03-31 00:00:00	The Department of Housing Preservation and Develop...
0	91	APARTMENT	542	BEDROOM	4	IMMEDIATE EMERGENCY	28	PAINT/PLASTER	197	CEILING	2520	COLLAPSING OR FALLING	2	CLOSE	2014-07-29 00:00:00	The Department of Housing Preservation and Develop...
0	91	APARTMENT	542	BEDROOM	3	NON EMERGENCY	28	PAINT/PLASTER	198	WALL	1364	BULGING/HOLE/CRACKED	2	CLOSE	2014-07-29 00:00:00	The Department of Housing Preservation and Develop...
0	91	APARTMENT	543	ENTIRE APARTMENT	1	EMERGENCY	59	HEATHOT WATER	348	APARTMENT ONLY	2633	NO HEAT AND NO HOT WATER	2	CLOSE	2015-03-06 00:00:00	The Department of Housing Preservation and Develop...
0	91	APARTMENT	543	ENTIRE APARTMENT	4	IMMEDIATE EMERGENCY	10	ELECTRIC	341	POWER OUTAGE	2678	ENTIRE APARTMENT	2	CLOSE	2014-08-08 00:00:00	The Department of Housing Preservation and Develop...
0	91	APARTMENT	543	ENTIRE APARTMENT	1	EMERGENCY	10	ELECTRIC	71	WIRING	2461	ILLEGAL	2	CLOSE	2014-08-08 00:00:00	The Department of Housing Preservation and Develop...
0	91	APARTMENT	541	BATHROOM	1	EMERGENCY	11	GENERAL	73	CABINET	680	FALLING OFF WALL	2	CLOSE	2014-08-20 00:00:00	The Department of Housing Preservation and Develop...
0	91	APARTMENT	541	BATHROOM	3	NON EMERGENCY	65	WATER LEAK	381	SLOW LEAK	2631	AT WALL OR CEILING	2	CLOSE	2014-10-24 00:00:00	The Department of Housing Preservation and Develop...
0	91	APARTMENT	546	KITCHEN	1	EMERGENCY	9	PLUMBING	65	BASIN/SINK	2534	PIPE LEAKING	2	CLOSE	2015-07-16 00:00:00	The Department of Housing Preservation and Develop...
0	91	APARTMENT	543	ENTIRE APARTMENT	3	NON EMERGENCY	63	UNSANITARY CONDITION	376	PESTS	2621	MICE	2	CLOSE	2017-03-16 00:00:00	The Department of Housing Preservation and Develop...
0	91	APARTMENT	541	BATHROOM	1	EMERGENCY	9	PLUMBING	66	TOILET	631	BOWL LOOSE OR WOBBLY	2	CLOSE	2017-03-03 00:00:00	The Department of Housing Preservation and Develop...
0	91	APARTMENT	541	BATHROOM	3	NON EMERGENCY	56	DOOR/WINDOW	334	DOOR FRAME	2668	FRAME BROKEN	2	CLOSE	2014-08-20 00:00:00	The Department of Housing Preservation and Develop...

Figura 3 - ComplainID contém apenas zeros

Para solucionar isto foi efetuado um *drop* da coluna “ComplainID” e criada novamente, mas definida como chave primária e como um valor que irá incrementar a cada registo, bloco 6. A execução deste comando demorou cerca de 30 segundos.

```
ALTER TABLE Complain
DROP COLUMN ComplainID;

ALTER TABLE Complain
ADD COLUMN ComplainID int(1) unsigned PRIMARY KEY AUTO_INCREMENT;
```

Bloco 7 - Drop da coluna e criação da coluna com auto incremento

Modelo Relacional

Esquema da Base de dados

O esquema da base de dados representa a configuração lógica de toda ou parte de uma base de dados relacional (Lucid, s.d.). Normalmente o esquema da base de dados é criado

por um designer de base de dados para ajudar os programadores a construir a base de dados.

Para a design desta base de dados foi utilizado um website, <http://draw.io>, que é bastante útil pois traz opções para design de base de dados relacionais e traz funcionalidades para interação entre as tabelas com chaves primárias, PK, e chaves estrangeiras, FK.

O design da base de dados pode ser observado na figura 3.

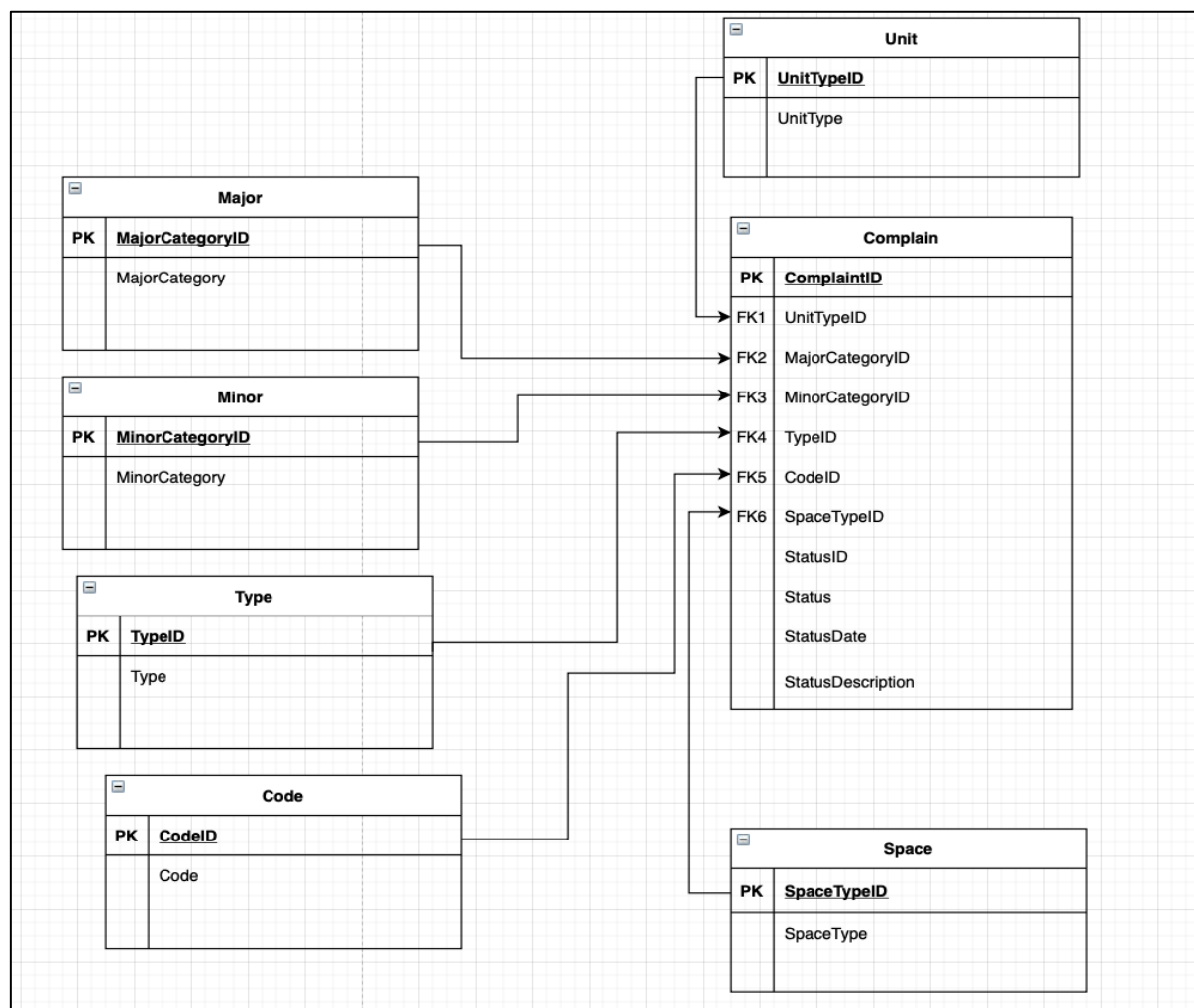


Figura 4 - Design da base de dados

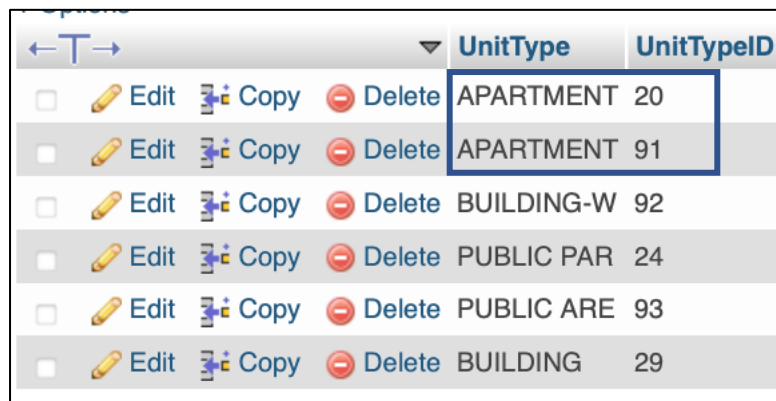
```
CREATE TABLE Type(TypeID int, Type VARCHAR(100));
CREATE TABLE Unit(UnitTypeID int, UnitType VARCHAR(100));
CREATE TABLE Space(SpaceTypeID int, SpaceType VARCHAR(100));
CREATE TABLE Major(MajorCategoryID int, MajorCategory VARCHAR(100));
CREATE TABLE Minor(MinorCategoryID int, MinorCategory VARCHAR(100));
CREATE TABLE Code(CodeID int, Code VARCHAR(100));
```

Bloco 8 – Criação das tabelas

Tabela Unit

A tabela Unit tem 2 colunas, UnitTypeID e UnitType. UnitTypeID é a chave primária desta tabela é um código para o UnitType que se trata do local mais genérico de onde ocorreu o incidente, *APARTMENT*, *BUILDING-W*, *PUBLIC PAR*, *PUBLIC ARE* e *BUILDING*.

Reparou-se que APARTMENT possuía 2 valores como UnitTypeID, figura 7.



	UnitType	UnitTypeID
<input type="checkbox"/> Edit Copy Delete	APARTMENT	20
<input type="checkbox"/> Edit Copy Delete	APARTMENT	91
<input type="checkbox"/> Edit Copy Delete	BUILDING-W	92
<input type="checkbox"/> Edit Copy Delete	PUBLIC PAR	24
<input type="checkbox"/> Edit Copy Delete	PUBLIC ARE	93
<input type="checkbox"/> Edit Copy Delete	BUILDING	29

Figura 5 – Valores duplicados

Isto é considerado uma incoerência e então foi escolhido um dos valores e teve de se substituir o outro valor por este mesmo com o comando no bloco 8.

```
UPDATE Complain
SET UnitTypeID = 91
WHERE UnitTypeID = 20;
```

Bloco 9 – Correção dos dados

Após esta correção pode-se avançar para o próximo passo que se trata da inserção dos valores na tabela Unit, bloco 9.

```
INSERT INTO Unit(UnitType, UnitTypeID)
SELECT DISTINCT(UnitType), UnitTypeID
FROM Complain;
```

Bloco 10 – Inserção dos dados na tabela Unit

Tabela Space

A tabela Space tem 2 colunas, SpaceTypeID e SpaceType. SpaceTypeID é a chave primária desta tabela é um código para o SpaceType que se trata do local mais específico de onde ocorreu o incidente, *ENTIRE APARTMENT*, *BEDROOM*, *BATHROOM*, etc.

Reparou-se que *ENTIRE APARTMENT* possuía 2 valores como UnitTypeID, figura 7.

				SpaceTypeID	SpaceType
<input type="checkbox"/>	Edit	Copy	Delete	68	ENTIRE APARTMENT
<input type="checkbox"/>	Edit	Copy	Delete	545	ENTRANCE/FOYER
<input type="checkbox"/>	Edit	Copy	Delete	543	ENTIRE APARTMENT
<input type="checkbox"/>	Edit	Copy	Delete	542	BEDROOM
<input type="checkbox"/>	Edit	Copy	Delete	541	BATHROOM
<input type="checkbox"/>	Edit	Copy	Delete	546	KITCHEN
<input type="checkbox"/>	Edit	Copy	Delete	550	BUILDING-WIDE
<input type="checkbox"/>	Edit	Copy	Delete	548	OTHER ROOM/AREA
<input type="checkbox"/>	Edit	Copy	Delete	547	LIVING ROOM
<input type="checkbox"/>	Edit	Copy	Delete	85	PUBLIC HALL
<input type="checkbox"/>	Edit	Copy	Delete	556	BUILDING ENTRANCE
<input type="checkbox"/>	Edit	Copy	Delete	566	HALLWAY
<input type="checkbox"/>	Edit	Copy	Delete	549	PRIVATE HALL
<input type="checkbox"/>	Edit	Copy	Delete	553	BASEMENT
<input type="checkbox"/>	Edit	Copy	Delete	574	STAIRS
<input type="checkbox"/>	Edit	Copy	Delete	573	ROOF
<input type="checkbox"/>	Edit	Copy	Delete	569	LOBBY
<input type="checkbox"/>	Edit	Copy	Delete	577	YARD
<input type="checkbox"/>	Edit	Copy	Delete	571	OTHER ROOM/AREA
<input type="checkbox"/>	Edit	Copy	Delete	560	COURTYARD
<input type="checkbox"/>	Edit	Copy	Delete	187	BASEMENT
<input type="checkbox"/>	Edit	Copy	Delete	81	LOBBY
<input type="checkbox"/>	Edit	Copy	Delete	78	FRONT AREAWAY
<input type="checkbox"/>	Edit	Copy	Delete	558	COMPACTOR CLOSET
<input type="checkbox"/>	Edit	Copy	Delete	562	FIRE ESCAPE/EGRESS

Figure 6 -Entire Apartment possui duas chaves diferentes

Por essa razão foi efetuado o mesmo procedimento que na tabela anterior, bloco 10.

```
UPDATE Complain
SET SpaceTypeID = 68
WHERE SpaceTypeID = 543;
```

Bloco 11 – Correção dos dados

Após esta correção pode-se avançar para o próximo passo que se trata da inserção dos valores na tabela Space, bloco 11.

```
INSERT INTO Space(SpaceTypeID, SpaceType)
SELECT DISTINCT(SpaceTypeID), SpaceType
FROM Complain;
```

Bloco 12 – Inserção dos dados na tabela Space

Tabela Type

A tabela Type tem 2 colunas, TypeID e Type. TypeID é a chave primária desta tabela é um código para o Type pode ser *Emergency*, *Immediate Emergency*, *Non Emergency* ou *Hazardous*.

Reparou-se que *Emergency* possuía 2 valores como TypeID, figura 9.

TypeID	Type
1	EMERGENCY
4	IMMEDIATE EMERGENCY
3	NON EMERGENCY
9	EMERGENCY
2	HAZARDOUS

Figura 7 – Valores TypeID e Type

Por esta razão foi utilizado um comando de update para terem o mesmo TypeID, bloco 13.

```
UPDATE Complain
SET TypeID = 1
WHERE TypeID = 9;
```

Bloco 13 - Correção de incoerências

Por fim serão inseridos os dados na tabela Type com o comando no bloco 14.

```
INSERT INTO Space(SpaceTypeID, SpaceType)
SELECT DISTINCT(SpaceTypeID), SpaceType
FROM Complain;
```

Bloco 13 – Inserção dos dados na tabela Type

Tabela Code

A tabela Code é composta por 2 colunas: CodeID e Code, sendo CodeID a Chave Primária. Esta mesma coluna é o código que permite identificar o problema em si, para ser reportado.

Ou seja, Code pode ser *Apartment Entrance*, *Hole*, etc.

Foi feita análise atenciosa a estas colunas, figura z.

Code	CodeID
*APARTMENT ENTRANCE	903
*BUILDING / VESTIBULE ENTRANCE	2478
*BUILDING ENTRANCE	905
*DOOR FRAME	2474
*EXPOSED	2480
*FAUCET IS BROKEN	2551
*FLOOR COVERING TORN	742
*HOLE	744
*HOLE	745
*HOLE	2484
*MECHANICAL NOT WORKING	2496
*MIRROR IN CAR	2452
*MISSING	2620
*TO ROOF	2477
*TO ROOM WITHIN APARTMENT	714
*THREAD STEP LOOSE	776
ACCUMULATION	2505
ACCUMULATION	2813
AFFECTING STRUCTURAL STABILITY	2830
AIR VALVE BROKEN OR MISSING	2542
APARTMENT CREATED IN CELLAR/ BASEMENT/ ATTIC	830
APARTMENT DOOR BROKEN OR DEFECTIVE	2472
AT WALL OR CEILING	2828
AT WALL OR CEILING	2831
BALCONY/ BASKET REMOVED	2480

Figura 8 – Valores duplicados

Após essa análise notou-se que havia valores repetidos e tinham de ser corrigidos.

Para isso foi utilizado o código do bloco 13.

```

UPDATE Complain
SET CodeID = 744
WHERE CodeID = 2484;

UPDATE Complain
SET CodeID = 744
WHERE CodeID = 745;

UPDATE Complain
SET CodeID = 2505
WHERE CodeID = 2813;

UPDATE Complain
SET CodeID = 2828
WHERE CodeID = 2831;

UPDATE Complain
SET CodeID = 745
WHERE CodeID = 744 and Code = "HOLE, NON-WALKING AREA";

DELETE FROM Complain
WHERE CodeID = 1389 and Code = "COLLAPSING/FALLING ";

```

Bloco 15 - Correção dos valores

Procedeu-se ao próximo passo, transferir os dados para a tabela Code, bloco 14.

```
INSERT INTO Code(CodeID, Code)
SELECT DISTINCT(CodeID), Code
FROM Complain;
```

Bloco 16 – Inserção dados na tabela Code

Tabela Major

A tabela Major tem 2 colunas, MajorCategoryID e MajorCategory. MajorCategoryID é a chave primária desta tabela, é um código para a MajorCategory que se trata da categoria geral que envolve o incidente, por exemplo, HEAT/HOT WATER, DOOR/WINDOW, etc.

Observando os dados e vendo que estes estavam todos coerentes, figura x, avançou-se assim à inserção dos valores na tabela Major, bloco 15.

MajorCategoryID	MajorCategory
59	HEAT/HOT WATER
56	DOOR/WINDOW
28	PAINT/PLASTER
10	ELECTRIC
11	GENERAL
65	WATER LEAK
9	PLUMBING
63	UNSANITARY CONDITION
58	FLOORING/STAIRS
8	APPLIANCE
13	NONCONST
61	OUTSIDE BUILDING
62	SAFETY
12	HEATING
57	ELEVATOR
27	CONSTRUCTION

Figura 9 – Verificação dos dados

```
INSERT INTO Major(MajorCategoryID, MajorCategory)
SELECT DISTINCT(MajorCategoryID), MajorCategory
FROM Complain;
```

Bloco 17 – Inserção dos dados na tabela Major

Tabela Minor

A tabela Minor tem 2 colunas, MinorCategoryID e MinorCategory. MinorCategoryID é a chave primária desta tabela, é um código para a MinorCategory que se trata da localização onde envolve o incidente, por exemplo, ENTIRE BUILDING, DOOR, WINDOW FRAME, etc.

Observando os dados e vendo que estes estavam todos coerentes, figura y, avançou-se assim para a inserção dos valores na tabela Minor, bloco 16.

MinorCategoryID	MinorCategory
349	ENTIRE BUILDING
333	DOOR
337	WINDOW FRAME
197	CEILING
198	WALL
348	APARTMENT ONLY
341	POWER OUTAGE
71	WIRING
73	CABINET
381	SLOW LEAK
65	BASIN/SINK
376	PESTS
66	TOILET
334	DOOR FRAME
63	BATHTUB/SHOWER
343	FLOOR
59	ELECTRIC/GAS RANGE
61	REFRIGERATOR
380	HEAVY FLOW
101	RUBBISH
345	BELL/BUZZER/INTERCOM
68	WATER SUPPLY
375	MOLD
81	COOKING GAS
379	DAMP SPOT

Figura 10 - Dados coerentes Minor

```
INSERT INTO Minor(MinorCategoryID, MinorCategory)
SELECT DISTINCT(MinorCategoryID), MinorCategory
FROM Complain;
```

Bloco 18 - Inserção dos dados na tabela Minor

Tabela Complain

A tabela central vai ser a tabela Complain.

Nesta tabela iremos ter todas as chaves primarias das outras tabelas e mais umas colunas de status.

Esta tabela contém todos os dados que temos passado para as outras tabelas e agora podemos dar drop nas colunas que não são chaves primárias nas outras tabelas, ou seja, UnitType, SpaceType, Type, MajorCategory, MinorCategory e Code.

Para efetuar o drop a estas colunas foi usado o código do bloco 17.

```
ALTER TABLE Complain
DROP COLUMN UnitType,
DROP COLUMN SpaceType,
DROP COLUMN Type,
DROP COLUMN MajorCategory,
DROP COLUMN MinorCategory,
DROP COLUMN Code;
```

Bloco 19 – Drop das colunas

Chaves Primárias e Estrangeiras

Após a obtenção das tabelas e de ter as incoerências corrigidas pode-se avançar para a criação das chaves primárias e estrangeiras.

Para isto foi utilizado o código que se encontra no bloco 18.

```
ALTER TABLE Code
ADD CONSTRAINT CodeID
PRIMARY KEY (CodeID);

ALTER TABLE Unit
ADD CONSTRAINT UnitTypeID
PRIMARY KEY (UnitTypeID);

ALTER TABLE Space
ADD CONSTRAINT SpaceTypeID
PRIMARY KEY (SpaceTypeID);

ALTER TABLE Type
ADD CONSTRAINT TypeID
PRIMARY KEY (TypeID);

ALTER TABLE Major
ADD CONSTRAINT MajorCategoryID
PRIMARY KEY (MajorCategoryID);

ALTER TABLE Minor
ADD CONSTRAINT MinorCategoryID
PRIMARY KEY (MinorCategoryID);
```

Bloco 20 - Definição das chaves primárias

Após a definição das chaves primárias é necessário a definição das chaves estrangeiras para termos as tabelas ligadas. A definição das chaves estrangeiras pode ser vista no bloco 19.

```
ALTER TABLE Complain
ADD FOREIGN KEY (CodeID) REFERENCES Code(CodeID);

ALTER TABLE Complain
ADD FOREIGN KEY (UnitTypeID) REFERENCES Unit(UnitTypeID);

ALTER TABLE Complain
ADD FOREIGN KEY (SpaceTypeID) REFERENCES Space(SpaceTypeID);

ALTER TABLE Complain
ADD FOREIGN KEY (TypeID) REFERENCES Type(TypeID);

ALTER TABLE Complain
ADD FOREIGN KEY (MajorCategoryID) REFERENCES Major(MajorCategoryID);

ALTER TABLE Complain
ADD FOREIGN KEY (MinorCategoryID) REFERENCES Minor(MinorCategoryID);
```

Bloco 21 – Definição das chaves estrangeiras

Após a definição das chaves podemos visualizar o produto final na secção de designer do phpmyadmin que é o esquema da base de dados, figura x.

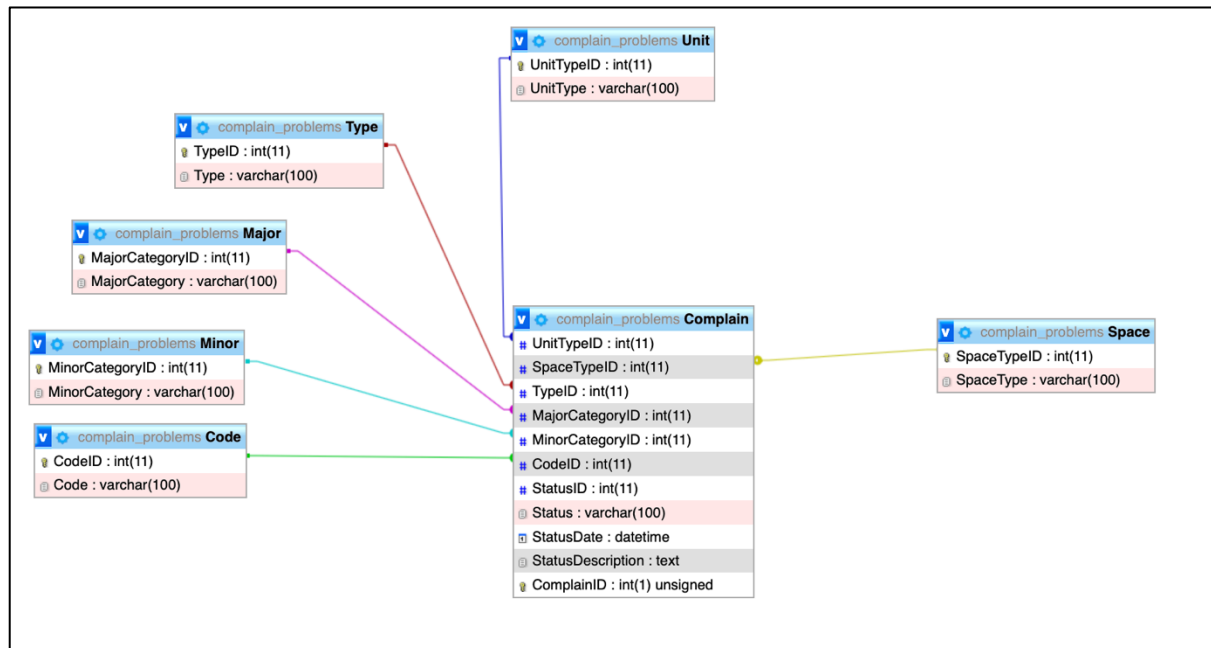


Figura 11 - Esquema da base de dados

Bibliografia

Tol, S. (2021, Jan 14). Retrieved from DZone: <https://dzone.com/articles/relational-vs-nosql-databases-and-rdbms-db-to-nosq>

(n.d.). Retrieved from Lucid: <https://www.lucidchart.com/pages/database-diagram/database-schema>