



**UNIVERSIDADE DE SÃO PAULO – USP SÃO CARLOS**  
**INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO**

**Segunda Avaliação (Remota)**  
**SCC240 – Bases de Dados**

Aluno: Sidnei Gazola Junior      Número USP: 9378888  
Responsável: Caetano Traina Junior

São Carlos – SP  
2020

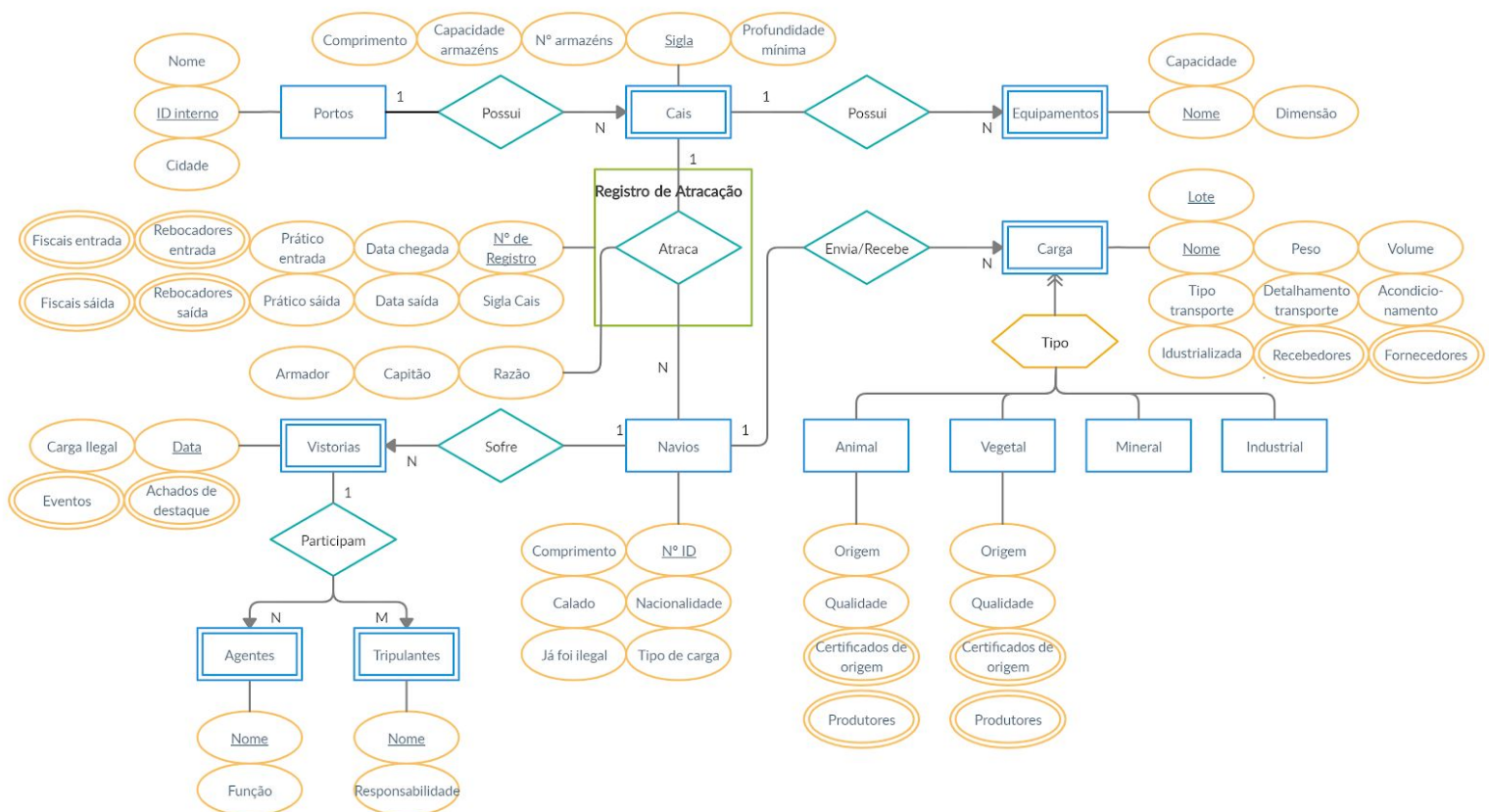
# SUMÁRIO

<b>EXERCÍCIO 1</b>	<b>3</b>
1.a)	3
1.b)	6
1.c.1)	7
1.c.2)	8
<b>EXERCÍCIO 2</b>	<b>9</b>
2.a)	9
2.b)	9
2.c)	10
2.d)	11
2.e)	12
<b>EXERCÍCIO 3</b>	<b>13</b>
3.a)	13
3.b)	15
3.c)	16
<b>EXERCÍCIO 4</b>	<b>17</b>
4.c)	17
4.d)	18
4.d.a)	18
4.d.b)	18
4.d.c)	18

# EXERCÍCIO 1

## 1.a)

A seguir o diagrama de entidades e relacionamentos estendido, e sua documentação.



## Documentação do diagrama

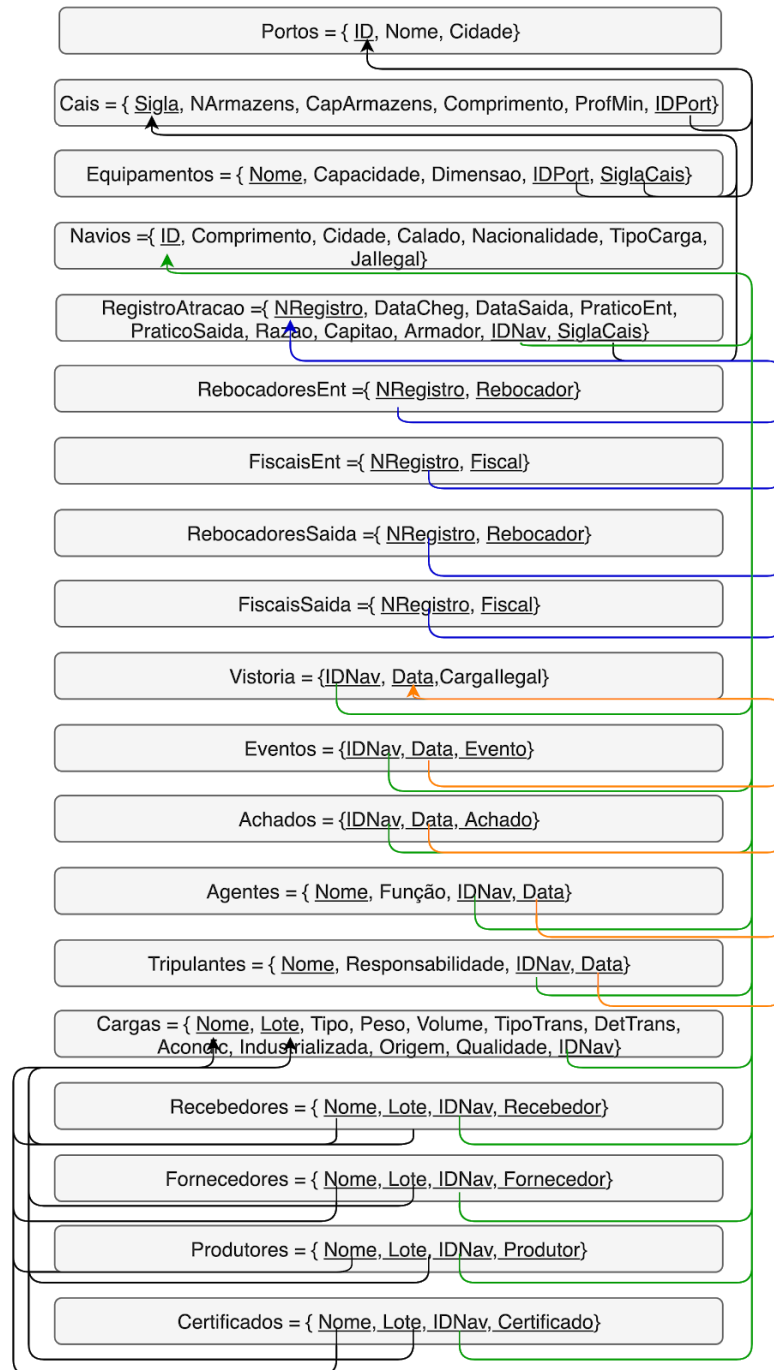
Entidades	Atributos	Tipo	Descrição
Portos	ID Interno	Numérico	ID interno do porto.
	Nome	Texto	Nome do porto.
	Cidade	Texto	Cidade do porto.
Cais	Sigla	Texto	Cada cais é identificado por sua sigla no porto.
	Nº armazéns	Numérico	Número de armazéns do cais.
	Capacidade armazéns	Numérico	Capacidade de armazéns do cais.
	Comprimento	Texto	Comprimento do cais.
	Profundidade mínima	Numérico	Profundidade mínima do cais.
Equipamentos	Nome	Texto	Nome dos equipamentos de movimentação de carga.
	Dimensão	Numérico	Dimensão do equipamento de movimentação de carga.
	Capacidade	Numérico	Capacidade dos equipamentos de movimentação de carga.
Navios	Nº ID	Numérico	Número ID identificador do navio. (Número IMO)
	Nacionalidade	Texto	Nacionalidade do navio.
	Tipo de carga	Texto	Tipo de carga transportada.
	Comprimento	Numérico	Comprimento do navio.
	Calado	Numérico	Calado do navio
	Já foi ilegal	Binário	Identifica se o Navio já foi ilegal (1 já foi ilegal, 0 não foi ilegal).
Carga	Nome	Texto	Número do lote identificador da carga.
	Lote	Numérico	Nome identificador da carga.
	Peso	Numérico	Peso da carga.
	Volume	Texto	Volume da carga.
	Tipo transporte	Texto	Forma com que a carga foi transportada (pode ser transportada a granel, paletizada ou containerizada).
	Detalhamento transporte	Texto	Dados específicos de como é transportada, se o tipo de transporte foi container terá os containers usados, se foi paletes, terá os paletes usados, se foi a granel terá a densidade a granel.
	Acondicionamento	Texto	Acondicionamento da carga.
	Idustrializada	Binário	Variável binária. Identifica se a carga é industrializada ou in natura (1 industrializada, 0 in natura).
	Recebedores	Texto	Atributo multivalorado. Recebe a lista de recebedores da carga.
	Fornecedores	Texto	Atributo multivalorado. Recebe a lista de Fornecedores da carga.
Animal	Tipo	Texto	Tipo da carga, pode ser de origem animal, vegetal, mineral ou industrial.
	Origem	Texto	Origem da carga do tipo animal.
	Produtores	Texto	Atributo multivalorado. Recebe a lista de produtores da carga do tipo animal.
	Certificados de origem	Texto	Atributo multivalorado. Recebe a lista de certificados de origem da carga do tipo animal.
	Qualidade	Texto	Origem da carga do tipo animal.
Vegetal	Origem	Texto	Origem da carga do tipo vegetal.
	Produtores	Texto	Atributo multivalorado. Recebe a lista de produtores da carga do tipo vegetal.
	Certificados de origem	Texto	Atributo multivalorado. Recebe a lista de certificados de origem da carga do tipo vegetal.
	Qualidade	Texto	Origem da carga do tipo vegetal.
Mineral	-	Entidade	Sub entidade da carga, do tipo mineral.
Industrial	-	Entidade	Sub entidade da carga, do tipo industrial.

<b>Vistoria</b>	<b>Data</b>	Data	Atributo composto do tipo data. Recebe a data da realização da vistoria.
	<b>Eventos</b>	Texto	Registra os achados de destaque na vistoria.
	<b>Achados de destaque</b>	Texto	Registra os eventos de destaque na vistoria.
	<b>Carga Ilegal</b>	Binário	Variável binária. Identifica se a vistoria encontrou carga ilegal ou não no navio (1 se encontrou, 0 se não encontrou).
<b>Agentes</b>	<b>Nome</b>	Texto	Agentes que participaram da vistoria.
	<b>Função</b>	Texto	Funções dos agentes que participaram da vistoria.
<b>Tripulantes</b>	<b>Nome</b>	Texto	Registra a responsabilidade dos tripulantes que interagiram na vistoria.
	<b>Responsabilidade</b>	Texto	Registra quais tripulantes interagiram na vistoria.
<b>Atraca</b>	<b>Razão</b>	Texto	Registra a razão da atracação pode ser carga ou descarga de mercadorias, manutenção, atendimento médico à população ou abastecimento.
	<b>Capitão</b>	Texto	Registra o capitão do navio no momento em que um navio atraca.
	<b>Armador</b>	Texto	Registra o armador do navio, que é uma empresa que organiza o transporte marítimo de cargas em rotas locais ou internacionais, operando os navios e movimentando tipos específicos de mercadorias entre os portos.
	<b>Nº do registro</b>	Numérico	Sempre que um navio chega, ele é atracado num cais que atenda às necessidades da carga e do tipo de navio, o que é registrado nesse Registro de Atracação, identificado por um Numero de Registro, registrado nesse atributo.
<b>Registro de atracação</b>	<b>Sigla Cais</b>	Texto	Registra a sigla do cais em que o Navio atracou.
	<b>Data chegada</b>	Data	Atributo composto do tipo data. Registra a data em que o Navio atracou.
	<b>Prático entrada</b>	Texto	Registra quem foi o prático na entrada do navio no cais (o capitão do rebocador ou manobrista no estuário).
	<b>Rebocadores entrada</b>	Texto	Atributo multivalorado. Registra a lista dos rebocadores que conduziu o navio na entrada.
	<b>Fiscais entrada</b>	Texto	Atributo multivalorado. Registra a listas dos fiscais de cada manobra na entrada.
	<b>Data partida</b>	Data	Atributo composto do tipo data. Registra a data em que o Navio partiu.
	<b>Prático saída</b>	Texto	Registra quem foi o prático na saída do navio do cais (o capitão do rebocador ou manobrista no estuário).
	<b>Rebocadores saída</b>	Texto	Atributo multivalorado. Registra a lista dos rebocadores que conduziu o navio na saída .
	<b>Fiscais saída</b>	Texto	Atributo multivalorado. Registra a listas dos fiscais de cada manobra na saída .



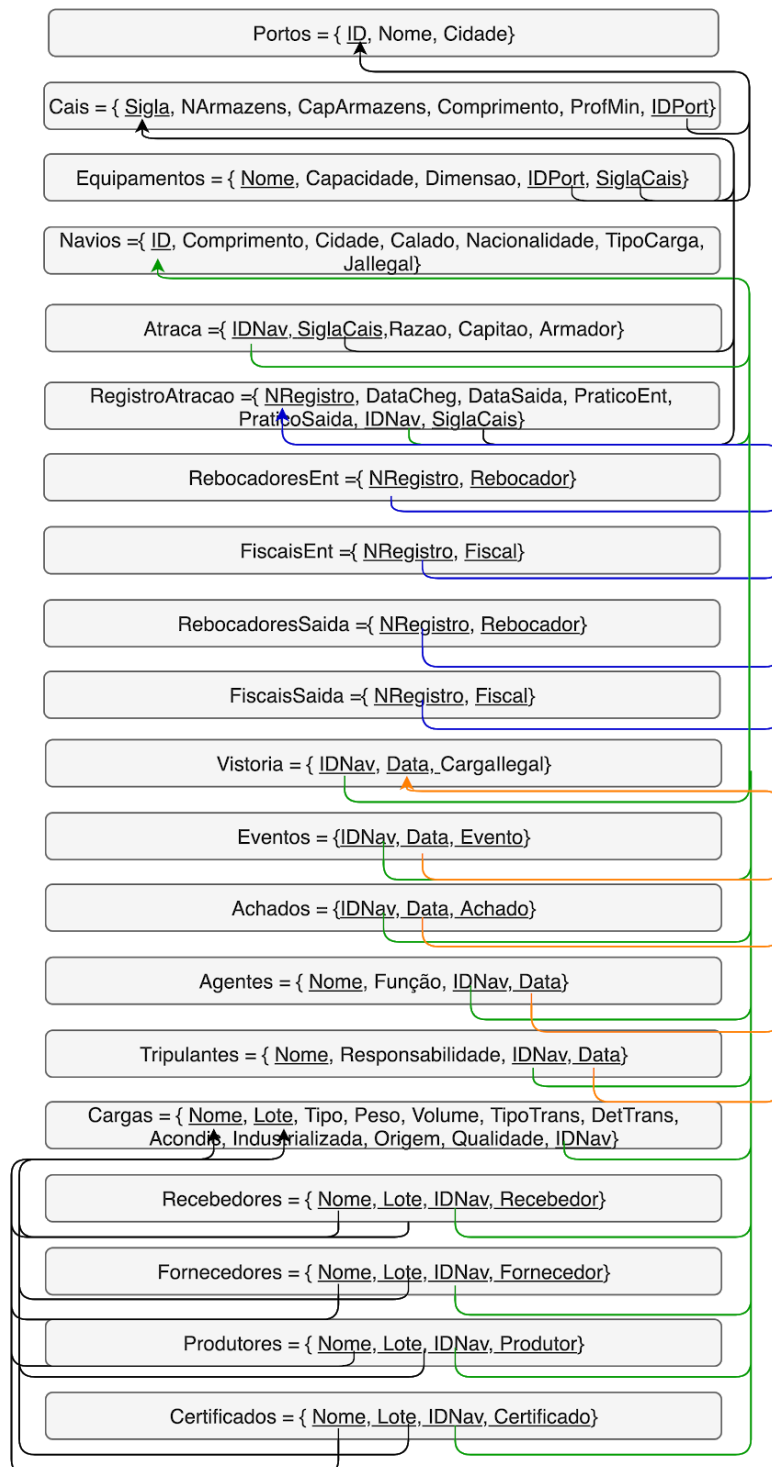
1.b)

Mapeamento para o modelo relacional.



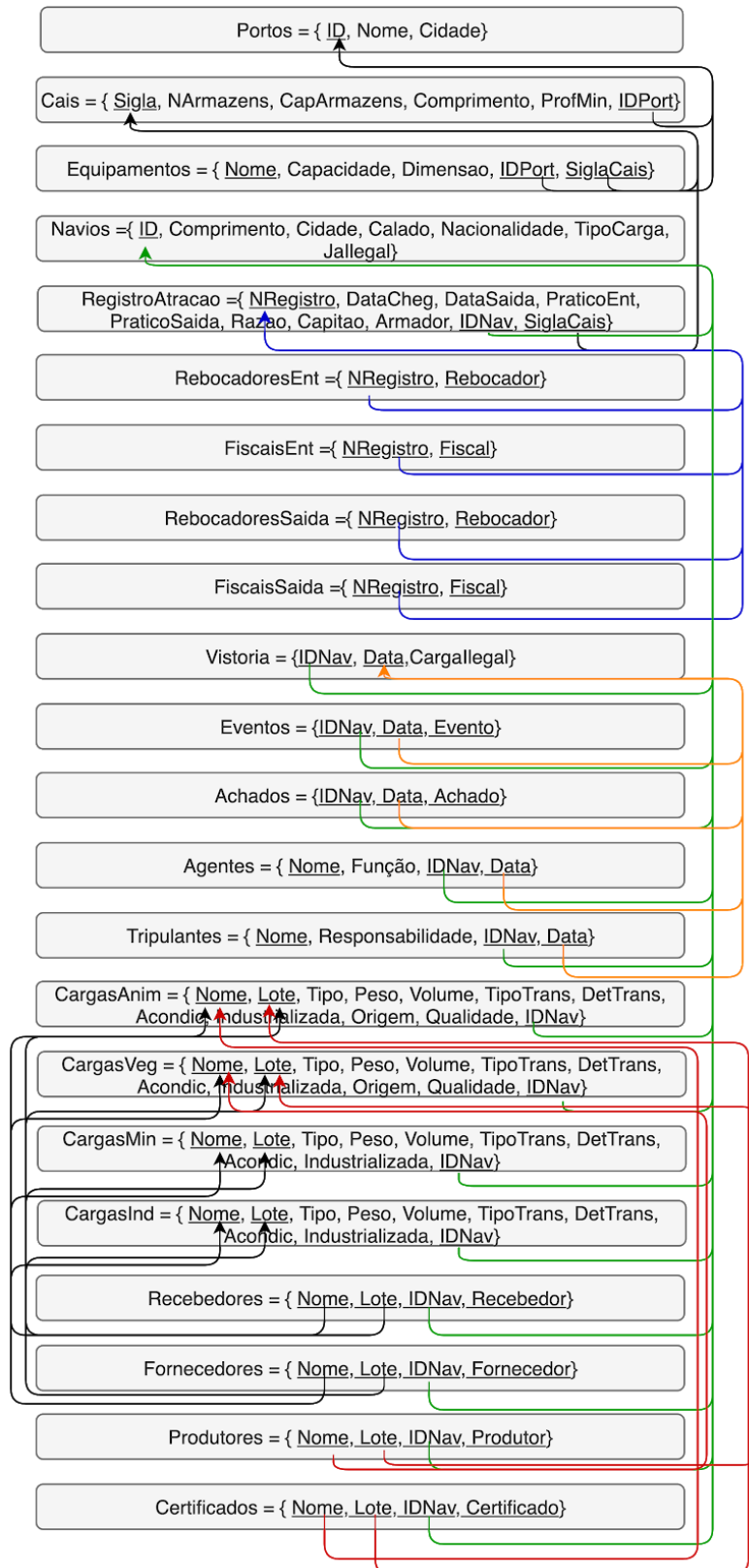
### 1.c.1)

A primeira alternativa de mapeamento, foi no registro de atracação, no mapeamento original, os atributos do Conjunto de Relacionamento Atraco foram mapeados na relação do Registro de Atracação, nesse outro fiz separado.



**1.c.2)**

A segunda alternativa de mapeamento, foi na relações de cargas, no primeiro mapeamento foi realizado somente uma relação para conter todas as caras, já nesta segunda, foram divididas por tipo.





## EXERCÍCIO 2

### 2.a)

A relação “Docas” não está na primeira forma normal, pois o atributo “elevadores” é multivalorado, o que pode ser resolvido da seguinte maneira:

Docas = {NumeroRegistro, DataChegada, ItemCarga, NomeCapitao, Tonelagem, Peso, Volume, Transportador}

Elevadores = {NumeroRegistro, DataChegada, ItemCarga, Elevador}

### 2.b)

- A relação “Aduana” na terceira forma normal ficaria:

Aduana = {NumeroRegistro, DataChegada, NomeRegistro, Nacionalidade, RegistroAnteriorCapitaoTipoCarga, NomeCapitão, Cargallegal, DataPartida}

RegistroAnteriorNavio = {NumeroRegistro, RegistroAnteriorNavio}

- A relação “Docas” na terceira forma normal ficaria:

Docas = {NumeroRegistro, DataChegada, ItemCarga, Peso, Volume, Transportador}

Elevadores = {NumeroRegistro, DataChegada, ItemCarga, Elevador}

Tonelagens = {NumeroRegistro, Tonelagem}

Capitão = {NumeroRegistro, DataChegada, NomeCapitao}

- A relação “Porto” na terceira forma normal ficaria:

Porto = {NumeroRegistro, DataChegada, Cais, RebocadorEntra, RebocadorSai, Fiscal}

Calados = {NumeroRegistro, Calado}

## 2.c)

Inconsistência devido à duplicidade de informações:

NumeroRegistro → RegistroAnteriorNavio

NumeroRegistro, DataChegada → NomeCapitao, TipoCarga, DataPartida, Cargallegal

### 1ª Inconsistência

Aduana={NumeroRegistro, DataChegada, NomeRegistro, Nacionalidade, TipoCarga, NomeCapitao, Cargallegal, DataPartida, RegistroAnteriorNavio, RegistroAnteriorCapitao} =

{<487, 13/11/2010, Animal,Brasil, Carlos, 0, 15/11/2010, 0, 1>,  
<487, 20/11/2010, Animal,Brasil, Carlos, 0, 15/11/2010, 1, 1>}

Está inconsistente pois para um mesmo NumeroRegistro temos dois RegistroAnteriorNavio diferentes. Ela é resolvida quando separamos o atributo RegistroAnteriorNavio, da seguinte forma:

RegistroAnteriorNavio = {NumeroRegistro, RegistroAnteriorNavio}

Dessa forma não é possível haver para um mesmo NumeroRegistro dois valores RegistroAnteriorNavio diferentes, pois só o NumeroRegistro é chave.

### 2ª Inconsistência

Aduana={NumeroRegistro, DataChegada, NomeRegistro, Nacionalidade, TipoCarga, NomeCapitao, Cargallegal, DataPartida, RegistroAnteriorNavio, RegistroAnteriorCapitao} =

{<487, 13/11/2010, Animal,Brasil, Carlos, 0, 15/11/2010, 0, 1>,  
<487, 13/11/2010, Animal,Peru, Bruno, 0, 15/11/2010, 0, 1>}

Está inconsistente pois para um mesmo NumeroRegistro e uma mesma DataChegada temos dois valores de NomeCapitao diferentes. Ela é resolvida quando separamos os atributos dependentes, da seguinte forma:

Navios = {NumeroRegistro, DataChegada, TipoCarga, NomeCapitão, Cargallegal, DataPartida}

Dessa forma não é possível haver para um mesmo NumeroRegistro e uma mesma DataChegada dois valores de NomeCapitao diferentes, pois só o NumeroRegistro e DataChegada são chaves.

## 2.d)

Perda de Dados devido à Alterações na Relação

- Relação “Aduana”:

NumeroRegistro → RegistroAnteriorNavio

Aduana={NumeroRegistro, DataChegada, NomeRegistro, Nacionalidade, TipoCarga, NomeCapitao, Cargallegal, DataPartida, RegistroAnteriorNavio, RegistroAnteriorCapitao} =

{<487, 13/11/2010, Animal,Brasil, Carlos, 0, 15/11/2010, 0, 1>, <887, 20/11/2005, Animal,Brasil, Carlos, 0, 25/11/2005, 1, 1>}

Se todas as tuplas com data de chegada do ano 2005, fossem excluídas, se perderia a informação do RegistroAnteriorNavio, desse Registro, o que pode ser antes de 2005. O que se resolve criando uma relação nova para guardar os registros anteriores, da seguinte maneira:

RegistroAnteriorNavio = {NumeroRegistro, RegistroAnteriorNavio}.

- Relação “Docas”:

NumeroRegistro → Tonelagem

Docas={NumeroRegistro, DataChegada, ItemCarga, NomeCapitao, Tonelagem, Peso, Volume, Transportador, Elevadores} =

{<487, 13/11/2010,Madeira,Brasil, Carlos, 86, 900, 20, Cargueiro, {Guincho,Guindaste}>, <887, 20/11/2005,Madeira,Brasil, José, 57, 580, 20, Cargueiro, {Guincho,Guindaste}>}

Se todas as tuplas do capitão José, fossem excluídas, se perderia a informação de quantas toneladas tiveram. O que se resolve criando uma relação, da seguinte maneira:

Tonelagens = {NumeroRegistro, Tonelagem}

- Relação “Porto”:

NumeroRegistro → Calado

Porto={NumeroRegistro, DataChegada, Calado, Cais, RebocadorEntra, RebocadorSai, Fiscal} =

{<487, 13/11/2010, 5, TN, Carlos, José, Bruno>, <887, 20/11/2005, 5, TN, Carlos, José, André>}

Se todas as tuplas do fiscal André, fossem excluídas, se perderia a informação do calado. O que se resolve criando uma relação, da seguinte maneira:

Calados = {NumeroRegistro, Calado}

2.e)

**select \* from Porto natural join Calados**

Por meio deste comando se faz junção da relação Porto com a relação Calados, por meio do atributo, chave nas duas relações NumeroRegistro.

Ficando assim:

Porto={NumeroRegistro, DataChegada, Cais, RebocadorEntra, RebocadorSai, Fiscal} =  
{<487, 13/11/2010, 5, Carlos, José, Bruno>,  
<887, 20/11/2005, 5, Carlos, José, André>}

Calados = {NumeroRegistro, Calado} =  
{<487, HY>,  
<887, KG>}

Porto={NumeroRegistro, DataChegada, Calado, Cais, RebocadorEntra, RebocadorSai, Fiscal} =  
{<487, 13/11/2010, 5, HY, Carlos, José, Bruno>,  
<887, 20/11/2005, 5, KG, Carlos, José, André>}

## EXERCÍCIO 3

### 3.a)

```
CREATE TABLESPACE TS1 LOCATION 'C:\Dados\TS1';
CREATE TABLESPACE TS2 LOCATION 'C:\Dados\TS2';
CREATE TABLESPACE TS3 LOCATION 'C:\Dados\TS3';
CREATE TABLESPACE TS4 LOCATION 'C:\Dados\TS4';

DROP TABLE IF EXISTS Movies CASCADE;
CREATE TABLE Movies(
    MovieId TEXT
        PRIMARY KEY USING INDEX TABLESPACE TS3,
    Title    TEXT,
    Year     INTEGER,
    RunTime  INTEGER,
    yt_score TEXT,
    tw_score TEXT
) TABLESPACE TS2;

COPY Movies
FROM 'C:\Dados\Movies.csv'
WITH (DELIMITER ',', NULL '', HEADER true, FORMAT CSV);

ALTER TABLE Movies DROP COLUMN yt_score;
ALTER TABLE Movies DROP COLUMN tw_score;

ALTER TABLE Movies ALTER COLUMN MovieId TYPE NUMERIC
    USING ((Substring(MovieId FROM '\d+'))::INT);

-----

DROP TABLE IF EXISTS Staff CASCADE;
CREATE TABLE Staff(
    PersonId TEXT
        PRIMARY KEY USING INDEX TABLESPACE TS2,
    Name     TEXT
)TABLESPACE TS3;

COPY Staff
FROM 'C:\Dados\Names.csv'
WITH (DELIMITER ',', NULL '', HEADER true, FORMAT CSV);

ALTER TABLE Staff ALTER COLUMN PersonId TYPE NUMERIC
    USING ((Substring(PersonId FROM '\d+'))::INT);
```

```

=====

DROP TABLE IF EXISTS Roles CASCADE;
CREATE TABLE Roles(
    MovieId TEXT,
    PersonId TEXT,
    Category TEXT,
    PRIMARY KEY (MovieId, PersonId) USING INDEX TABLESPACE TS4
)TABLESPACE TS1;

CREATE TEMP TABLE temporario
ON COMMIT DROP
AS
SELECT *
FROM Roles
WITH NO DATA;

COPY temporario
FROM 'C:\Dados\Principals.csv'
WITH (DELIMITER ',', NULL '', HEADER true, FORMAT CSV);

INSERT INTO Roles
SELECT * FROM temporario ON CONFLICT DO NOTHING;

ALTER TABLE Roles ALTER COLUMN PersonId TYPE NUMERIC
USING ((Substring(PersonId FROM '\d+'))::INT);
ALTER TABLE Roles ALTER COLUMN MovieId TYPE NUMERIC
USING ((Substring(MovieId FROM '\d+'))::INT);

```



### 3.b)

```
--1) todos indices desabilitados
SET enable_indexscan = False;
SET enable_indexonlyscan = False;
SET enable_bitmapscan = False;
explain analyse select staff.name
      from movies natural join roles natural join staff
      where movies.year =2015 and
            movies.title like '%Looking Glass%';

--2) com indices padroes habilitados
SET enable_indexscan = True;
SET enable_indexonlyscan = True;
SET enable_bitmapscan = True;
explain analyse select staff.name
      from movies natural join roles natural join staff
      where movies.year =2015 and
            movies.title like '%Looking Glass%';

--3) com outros indices criados habilitados
create index IdxMovieYear on movies(year) TABLESPACE TS3;
create index IdxMovieTitle on movies(title) TABLESPACE TS3;
explain analyse select staff.name
      from movies natural join roles natural join staff
      where movies.year =2015 and
            movies.title like '%Looking Glass%';
```

**Tempos de execução obtidos**

	1)	2)	3)
Planning Time	0.530 ms	0.466 ms	0.586 ms
Execution Time	1124.564 ms	25.101 ms	5.050 ms

Como esperado, conforme se adiciona mais índices, as buscas se tornam mais rápidas, como podemos observar pelos tempos obtidos, que foram decaindo conforme se adiciona mais índices.

### 3.c)

```
--1) todos indices desabilitados
SET enable_indexscan = False;
SET enable_indexonlyscan = False;
SET enable_bitmapscan = False;
explain analyse select movies.title
    from movies natural join roles natural join staff
    where staff.name='Frances Conroy' or
           staff.name='Edward Norton'
    group by movies.title
           having count(movies.title)>1;

--2) com indices padroes habilitados
SET enable_indexscan = True;
SET enable_indexonlyscan = True;
SET enable_bitmapscan = True;
explain analyse select movies.title
    from movies natural join roles natural join staff
    where staff.name='Frances Conroy' or
           staff.name='Edward Norton'
    group by movies.title
           having count(movies.title)>1;

--3) com outro indice adicional criado habilitado
create index IdxStaffName on staff(name) TABLESPACE TS2;
explain analyse select movies.title
    from movies natural join roles natural join staff
    where staff.name='Frances Conroy' or
           staff.name='Edward Norton'
    group by movies.title
           having count(movies.title)>1;
```

**Tempos de execução obtidos**

	1)	2)	3)
Planning Time	0.459 ms	0.473 ms	0.475 ms
Execution Time	1193.170 ms	933.691 ms	167.146 ms

Como esperado, conforme se adiciona mais índices, as buscas se tornam mais rápidas, como podemos observar pelos tempos obtidos, que foram decaindo conforme se adiciona mais índices. Neste caso fica mais perceptível a diferença de tempo com os índices adicionais criados, porém foi demorado para se criar o índice.

## EXERCÍCIO 4

### 4.c)

- Atomicidade

Essa propriedade garante que uma transação é executada por completo, se só for executada partes, tem que se abortar.

```
update banco set saldo = saldo -20 where conta=101;  
update banco set saldo = saldo +20 where conta=102;
```

Se essa transação for executada somente uma parte, uma conta perde o dinheiro mas a outra não ganha.

- Consistência

Essa propriedade garante que uma transação não pode entregar a base inconsistente.

```
INSERT INTO banco (conta) SELECT conta FROM banco WHERE Nome= José;
```

Esse comando é inconsistente pois adiciona uma conta que já existe no banco.

- Isolamento

Essa propriedade garante que uma transação não pode ver o efeito da outra até a outra ser executada por completo.

```
--T1  
update banco set saldo = saldo -20 where conta=101;  
--T2  
update banco set saldo = saldo +10 where conta=101;
```

Se uma transação interferir na outra sem ter terminado, vai gerar resultado inconsistentes na base.

- Durabilidade

Essa propriedade garante que uma transação não perderá sua validade, e durará até outra transação alterar os dados explicitamente, mesmo se houver uma pane no disco que armazena os dados.

```
update banco set saldo = saldo -20 where conta=101;  
update banco set saldo = saldo +20 where conta=102;
```

Se houver uma pane no disco e essa transação não restaurada, causará uma inconsistência na base.

#### **4.d)**

##### **4.d.a)**

O momento em que levaria o maior tempo de recuperação seria no último dia do mês às 1h59min. Assim teria que restaurado o dump (5min), mais 6 checkpoints (30s), considerando 30 dias em um mês, pois os checkpoint são feitos a cada 5 dias, mais a reexecução das transações do arquivo log desde o dia 25, o que dá 5 dias de transações ou seja em média  $1250 T1 + 250 T2$ . Portanto o tempo total de recuperação seria  $5\text{min} + 6 \cdot 30\text{s} = 8\text{min}$ , mais o tempo de reexecução das transações do arquivo log, que são em média  $1250 T1 + 250 T2$ .

##### **4.d.b)**

O momento em que levaria o menor tempo de recuperação seria no último dia do mês às 2h01min. Assim teria que restaurado apenas o dump (5min), mais 1 minuto de gravação no log, o que daria em média 1 transação  $T1 + 1$  transação  $T2$ . Portanto o tempo total de recuperação seria 5 minutos mais o tempo de recuperação do log o que daria aproximadamente 1 transação  $T1 + 1$  transação  $T2$ .

##### **4.d.c)**

Para recuperação da base no dia 11, às 11h, primeiramente teria que recuperar o arquivo dump (5min), e depois os logs do dia 5 e do dia 10 (30s cada), depois teria que reexecutar o arquivo log desde o último checkpoint (dia 10), então teriam que ser executadas em média 250 transação  $T1 + 50$  transação  $T2$ . Então todas as transações desde o último checkpoint teriam que ser reiniciadas.